A Mean Field Game Approach to Urban Drainage Systems Control: A Barcelona Case Study

Andrés Felipe Ramírez Jaime

Electronic Engineer (Bsc.)

Advisors

Dr. Carlos Ocampo Martínez
Dr. Nicanor Quijano Silva

Master in Automatic Control and Robotics
Escola Tècnica Superior d'Enginyeria Indistrial de Barcelona
Universitat Politècnica de Catalunya

A dissertation submitted for
the degree of Master in Automatic Control and Robotics

March 2016

Universitat Politècnica de Catalunya

Escola Tècnica Superior d'Enginyeria Indistrial de Barcelona

Title:

A Mean Field Game Approach to Urban Drainage Systems Control: A Barcelona Case Study

Master's Thesis made at:

Universitat Politècnica de Catalunya - Campus Barcelona

Carrer de Llorens i Artigas 4-6, 08028

Advisors:

Dr. Carlos Ocampo Martínez

Dr. Nicanor Quijano Silva

To Nobody

ABSTRACT

Urban drainage systems (UDSs) are complex large-scale systems that carry stormwater and wastewater throughout urban areas. During heavy rain scenarios, UDSs are not able to handle the amount of extra water that enters the network and flooding occurs. Usually, this might happen because the network is not being used efficiently, i.e., some structures remain underused while many others are overused. This thesis proposes a control methology based on mean field game theory and model predictive control that aims to efficiently use the existing network elements in order to minimize overflows and properly manage the water resource. The proposed controller is tested on a UDS located in the city of Barcelona, Spain, and is compared with a centralized MPC achieving similar results in terms of flooding minimization and wastewater treatement plant usage, but only using local information on non-centralized controllers and using less computation times.

Keywords: Urban Drainage Systems, Mean Field Games, Model Predictive Control, Hybrid Linear Delayed.

RESUMEN

Los sistemas de drenaje urbano son sistemas complejos y de gran escala que transportan tanto aguas negras como aguas de lluvia a través de las zonas urbanas. Durante eventos de precipitación muy fuertes, estos sistemas no son capaces de manejar la cantidad de agua adicional que entra y ocurren inundaciones indeseadas. Normalmente, esto pasa porque la red no está siendo utilizada eficientemente, i.e., algunas estructuras permanecen sub-utilizadas y otras se encuentran sobre-utilizadas. Esta tesis propone un esquema de control basado en la teoría de mean field games y de control predictivo que busca utilizar eficientemente la red para minimizar las inundaciones en cualquier escenario, y garantizar una utilización adecuada del recurso hídrico. El esquema propuesto es probado en la red de drenaje de la Riera Blanca en la ciudad de Barcelona y es comparado con una estrategia de control basada puramente en control predictivo, obteniendo resultados similares en terminos de inundación y utilización de plantas de tratamiento, pero utlizando únicamente una fracción del costo computacional.

Palabras clave: Sistema de Drenaje Urbano, Juegos de Campo Medio, Control Predictivo, Modelo Lineal Híbrido con Retardos.

RESUM

Sistemes de drenatge urbà (UDS) són sistemes complexos a gran escala que transporten aigües pluvials i residuals en les zones urbanes. Durant escenaris de fortes pluges, UDSs no són capaços de manejar la quantitat d'aigua addicional que entra a la xarxa i es produeix la inundació. En general, això pot succeir perquè la xarxa no està sent utilitzat de manera eficient, és a dir, algunes estructures segueixen sent infrautilitzades mentre que molts altres són usats en excés. En aquest treball es proposa una metodologia de control basat en la teoria de mean field games i control predictiu que pretén utilitzar de forma eficient els elements de xarxa existents per tal de minimitzar els desbordaments i gestionar adequadament els recursos hídrics. El controlador proposat es prova en un UDS situada a la ciutat de Barcelona, Espanya, i es compara amb un MPC centralitzat aconsegint resultats similars en termes de minimització de les inundacions i la utilització de la planta de tractament d'aigües residuals, però només utilitzant informació local en els controladors no centralitzats i l'ús de menys temps de càlcul.

Paraules clau: Sistemes de Drenatge Urbà, Jocs de Camp Mitjà, Control Predictiu, Model Híbrid Lineal amb Retards.

ACKNOWLEDGEMENT

Agradezco a Carlos y Nicanor por aguantarme, a Ye, Julián y Gerardo por escucharme y a Bernat por ayudarme.

Andrés Ramírez
Barcelona, March 2016

CONTENTS

List of Tables

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Motivation

Urban drainage systems (UDS) are complex networks of interconnected pipes and nodes that carry stormwater and wasterwater to wastewater treatment plants, which treats it and sends it to the enviroment [BD04]. In many cases, the design of these networks ends up being underdimensioned, because of rapid urbanization in cities and climate change scenarios, not being taken into account in early stages of the design process [WANON12]. For that reason, heavy flooding may appear in urban areas, and some serious sanitary problems may occur due to the unproper management of wastewater that comes out of the network into street level [BD04]. Some of the adressed solutions to that problem seek to do a restructuration to the hydraulic design of the network by adding storage elements throughout the system, so that the overflows are totally avoided [NADZ13]. Even though these solutions are quite effective, they are extremely expensive in both time and money.

The problem above could be solved, without having many large modifications being performed to the general design of the network, by using real-time control (RTC) techniques [GBGE+15]. These techniques seek to find a way to properly manage the active elements of the network, e.g., retention and redirection gates, in order to achieve an efficient management of the wastewater, and thus, assuring a minimization of overflows that may appear. Optimization-based control techniques have been the most widely used techniques in the literature to solve the problem of minimization of overflows in UDS. For instance, model predictive control (MPC) has been widely used to solve the problem [CQS+04, OM10], due to its flexibility in the selection of performance functions, constraints, and its multiple-inputs multiple-outputs capabilities

[Mac02]. However, many of the proposed techniques are based upon centralized schemes for the determination of control actions to be perfomed, which could derive into heavy computational burden problems [MMDLPC11] and cyber-security-related problems [CAS08].

In order to deal with some of the computational burden problems, aggregated models of the UDSs, e.g., the so-called virtual-tank (VT) model, are used to reduced the number of states of the system and, in term, the size of the optimization problems. These approaches are quite effective for many system, but deliver poor results when the network is heavly interconnected, i.e., when there is a large amount of connections among the pipes and nodes [JD14]. For that matter, there has been an increased interest in techniques that do not use aggregated models of the system, such as the one proposed in [JD14], where each element of the system is taken into consideration, without sacrificing computational time due to its linearity. Nonetheless, it is possible to encounter quite complex networks that require partitioning and decentralization, in order to guarantee suitable computation times for real-time applications.

For that reason, there has been an increased interest in studying distributed control techniques [CSMndlPnL13]. For instance, [BGORB$^+$15] propose a distributed control methodology based on population dynamics, that achieves an efficient use of the network and guarantees a minimization of flooding. However, on that methodology, local controllers are not able to consider proper cost functions, which can be problematic if there are multiple control goals such as, moving wastewater between a wastewater treatment plant (WWTP) and out of the network, while efficiently using the network and minimizing flooding. Moreover, the technique requires an aggregated model, which could derive into poor results, as stated before.

This thesis proposes a technique that aims to solve problems related to distributed information on local controllers, as well as problems related to the aggregation of large portions of the netwoks into single variables, by using a game-theoretic approach (i.e., dynamic games) combined with a hybrid linear delayed (HLD)-based MPC. Differential game (DG) theory [BO95] gives a natural extension of optimal control to scenarios with multiple controllers that are optimizing its own performance criteria [MSA14], and thus its framework is well suited for optimization-based non-centralized control applications. This type of games have been used in the literature to solve problems related to the formation control of mobile robots [Gu08], problems related to demand response in power grids [FEMRH15], and the control of surge tanks [FKV12]. This is due to the fact that DGs have the ability to consider multiple cost functions as well as non-centralized information on distributed controllers. As for the UDS control problem, it has been reported that these networks can be seen as partitioned systems that are being controlled by multiple local agents that interact with each other [BGORB$^+$15]. Hence, it is a suitable idea to apply the DGs framework to the control of UDS, where multiple local controllers act as players of a game where they interact with each other, in order to guarantee a proper operation of the network in terms of wastewater management.

It is quite important to point out that, even though DGs are quite useful for many non-centralized control application, they generally fail to succeed when the number of sub-systems, i.e., the number of agents, is large, because in order to compute the solution to a DG, it is required to solve a set coupuled partial differential equations (one per each agent in the game). Nonetheless, a novel tool called the mean field games (MFGs) [LL07, HCM03] allows to solve large scale DGs in which the number of agents tend to infinity. Hence, it still is suitable idea to use DG to solve non-centralized control problems.

The main contribution of this thesis is the design of a non-centralized control methodology based on large-scale DGs, i.e., a MFG, in the same spirit as in [NCMH13, BMA14], which seeks to determine the optimal behavior of each active element of the UDS by using a consensus-like algorithm, only using local information of the network. The proposed methodology has the advantages of optimization-based techniques used for the control of UDS, e.g., MPC, as well as the ability to have distributed information in the controllers. Moreover, since only local information is used, less data is needed, and thus less computational resources are involved in the computation of the control inputs. The proposed methodology is flexible enough that it allows to combine game-theoretic approaches with more traditional approaches, such as the MPC.

1.2 Thesis Objectives

- Study the mean field games and its applications to engineering problems.
- Study the relationship that exist between mean field games and predictive control.
- Design a control stretegy that combines mean field games and predictive control.
- Apply the proposed strategy to a real combined sewer system.

1.3 Outline of the Thesis

The remainder of this thesis is organized as follows. Chapter II presents a introductory background on dynamic game theory. It shows all the necesary tools to understand the ideas behind this thesis. Chapter III presents some preliminary results that show the main limitations behind differential games. Chapter IV presents the relationship that exists between the dynamic games and the UDSs. Therein, the concepts behind the agents and the enviroment are presented, as well as the different cost functions that the agents might minimize. Chapter IV also states the main problem to be solve in this thesis, as well as the required tools for a real-time implementation of the solution. Chapter V presentes the case study in which the approach is tested. It shows a portion of the UDS found in the city of Barcelona, Spain, called the Riera Blanca network. Chapter V also presents the main results obtained using the proposed approach, as well as a comparison between the proposed scheme and a more traditional tool for solving the problem. Finally, Chapter VI collects all the conclusions obtained after finishing the thesis.

Related Publications

Chapter III is entirely based on

A. RAMIREZ-JAIME, N. QUIJANO, C. OCAMPO-MARTINEZ. A Differential Game Approach to Urban Drainage System Control. *American Control Conference*, 2016. (*Accepted*)

Part I

State of the Art and Problem Statement

CHAPTER 2

BACKGROUND

This chapter aims to present the main concepts behind dynamic games. Since this thesis is concerned with this type of games, it is a suitable idea to introduce the reader to this notions. Only the main topics are presented, for a more detailed presentation, the reader should refer to a full text book regarding the area, e.g., [BO95].

2.1 Differential Games

Mean field games (MFGs) are a kind of differential games (DGs) in which a large number of agents are involved. Hence, it is convenient to first describe the main concepts behind DGs theory, in order to fully understand how the MFGs work.

Consider a traditional optimal control problem (OCP) of a dynamical system whose evolution is given by the following ordinary differential equation (ODE)

$$\dot{x}(t) = f(x(t), u(t), t), \qquad x(0) = x_0, \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state of the system, $u \in \mathbb{R}^m$ is the input (or control signal) of the system, and $t \in [0, T]$ is the time. The objective of the OCP is to determine a control signal u that optimizes the following performance criteria

$$J = \int_0^T g(x(t), u(t), t)dt + G(x(T), T), \tag{2.2}$$

where $g(\cdot, \cdot, \cdot)$ is known as the running cost, and $G(\cdot, \cdot)$ is known as the terminal cost. The

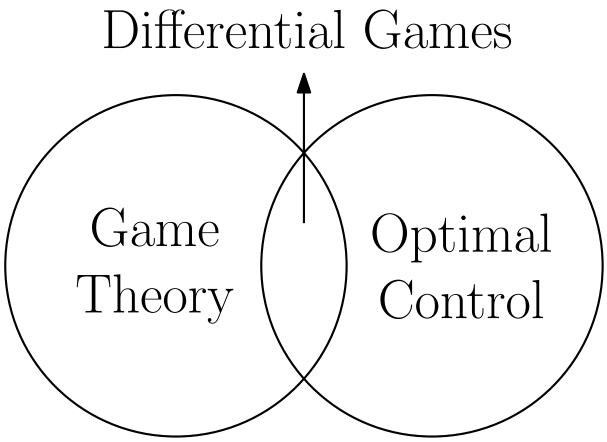


Figure 2.1: A DG can be defined as the relationship that exist between game theory and optimal control.

problem of optimizing (2.2) subject to (2.1) can be used to solve several known problems, e.g., moving a vehicle using the least amount of fuel, tracking a set-point in the least amount of time, among others.

DGs extend the idea behind OCPs to scenarios in which there are several independent control inputs altering the evolution of the system, which in turn are trying to optimize their own performance criteria. Therefore, a DG can be defined as the optimal control of a dynamical system that has $N \geq 2$ independent inputs; hence, each *agent* in the game manipulates a control input u_i of the system, which ultimately determines its evolution. Nonetheless, each agent is optimizing a performance criteria similar to (2.2), which may depend on the actions of other, as well as on the current state of the system (that is being alter by every agent), and this simlutaneous optimization is known as a game. Figure 2.1 shows the relationship between game theory and optimal control, i.e., the DGs.

2.1.1 Main Framework

Following the ideas from [Bre11], the main framework of the DGs is presented. Without loss of generality, the model is presented for the two-players case, but it can be augmented to any number of agents.

Consider a dynamical system in $x \in \mathbb{R}^n$, evolving in time according to

$$\dot{x}(t) = f(t, x, u_1, u_2), \qquad t \in [0, T], \;\; x(0) = x_0, \tag{2.3}$$

where $u_1(\cdot), u_2(\cdot)$ are the control signals or *strategies* implemented by the agents (also known as players). These strategies must satifiy

$$u_1(t) \in U_1, \qquad u_2(t) \in U_2, \tag{2.4}$$

for some given sets $U_1, U_2 \subseteq \mathbb{R}^n$.

Each agent in the game is optimizing a performance criteria given by

$$J_i(u_1, u_2) = \psi_i(x(T)) + \int_0^T L_i(t, x(t), u_1(t), u_2(t))dt, \qquad i \in \{1, 2\}, \tag{2.5}$$

where L_i are the running costs and ψ_i are the terminal costs. The functional (2.5) is a map that returns a scalar value for any pair of strategies u_1 and u_2 chosen by the agents, and thus, they must choose their actions so that their payoffs are optimized, taking into account the decisions of others.

In order to describe the general framework of a DG, it is necessary to describe the type of stretegy that a given agent is using. These stretegies depend on the information that each agent has available regarding the current state of the game. If a player does not have available the current state of the game, and only knows its initial condition, the strategies are *open-loop strategies*. Similarly, if an agent has available the current state of the game, the strategies are *closed-loop* or *feedback strategies*.

Finally, the following assumptions must be made in order to fully describe the game:

- Every agent in the game has knowledge of the evolution of the game, i.e., the function f.
- Every agent in the game has knowledge of the current time of the game, i.e., $t \in [0, T]$.
- Every agent in the game has available the initial state of the game, i.e., $x(0) = x_0 \in \mathbb{R}^n$.

In optimal control theory, the solution to an OCP refers to finding a control signal that optimizes a given performance criteria. However, notion of a solution in a DG might not be as apparent, since any solution of a game is associated to an *equilibrium* of the game. Traditionally in game theory, there can be several types of equilibria in a game, e.g., Nash, Pareto, or

Stackelberg. Nonetheless, for this thesis, only the Nash equilibrium is considered and thus the solution the of game is refered as the set of control signals that determine the Nash equilibrium. According to [Nas50], the Nash equilibrium of a N-player game, where u_i is the strategy that the i-th player is using, is defined as:

> "An equilibrium point is an N-tuple $\{u_1, u_2, ..., u_N\}$ such that each player's strategy optimizes his payoff if the strategies of others are held fixed. Thus each player's strategy is optimal againts those of the others."

Nash's definition can be written as follows (and assuming that each player is minimizing his cost functional):

$$J_i(u_1^*, u_2^*, \ldots, u_i^*, \ldots, u_N^*) \leq J_i(u_1^*, u_2^*, \ldots, u_i, \ldots, u_N^*), \tag{2.6}$$

where J_i is the performance criteria that the i-th player is minimizing, $\{u_1, u_2, \ldots, u_N\}$ are any set of admisible strategies, and $\{u_1^*, u_2^*, \ldots, u_N^*\}$ are the strategies that determined that Nash equilibrium of the game. Hence, any set of strategies that simultaneously satisfy the above inequalities, are a solution to the game, i.e., are Nash optimal. It is very important to point out that a Nash equilibrium may not be unique.

2.1.2 Open-loop Strategies

As it has been stated before, there are several types of strategies that the players in a DG can use. If the players have available the initial state of the game but do not have access to the current state, the strategies used by the players are called *open-loop strategies*. This implies that in order to solve the game, the agents must find two strategies u_1^* and u_2^* that simultaneously minimize their performance criteria, only using the initial condition $x(0) = x_0$.

Recalling the traditional OCP described by (2.1) and (2.2), it is possible to determine the optimal control signal $u^*(t)$ only using the initial state of the system, by solving the so-called canonical equations of optimal control [Kir12]. This canonical equations state that in order to solve the OCP it is require to solve the following system

$$
\boxed{\begin{aligned}
\dot{x}(t) &= f(x(t), u^*(t), t), \\
\dot{p}(t) &= -p(t)\frac{\partial f}{\partial x}(x(t), u^*(t), t) + \frac{\partial g}{\partial x}(x(t), u^*(t), t), \\
u^*(t) &= \arg\min_{u \in U} \Big\{ g(x(t), u^*(t), t) + p \cdot f(x(t), u(t), t) \Big\}, \\
x(0) &= x_0, \\
p(T) &= \nabla G(x(T)),
\end{aligned}} \tag{2.7}
$$

where $p \in \mathbb{R}^n$ is the co-state of the system, $u^* \in \mathbb{R}^m$ is the optimal control input, and $x(0), p(T) \in \mathbb{R}^n$ are the initial and terminal conditions for the state and co-state, respectively. This set of equations come from formulation the Euler-Lagrange equation for the cost functional (2.2) and setting the state equation (2.1) as the constraint.

In order to determine the open-loop solution to the DG, it is possible to use the same tools as for the OCP. However, since the goal is not to determine a single function but two control signals, the problem becomes more complex. For the DG scenario, in order to determine the set of strategies that characterized the solution of the game described by (2.3) and (2.5), it is required to solve the following set of canonical equations:

$$
\boxed{\begin{aligned}
\dot{x}(t) &= f(x(t), u_1^*(t), u_2^*(t), t), \\
\dot{p_1}(t) &= -p_1(t)\frac{\partial f}{\partial x}(x(t), u_1^*(t), u_2^*(t), t) - \frac{\partial L_1}{\partial x}(x(t), u_1^*(t), u_2^*(t), t), \\
u_1^*(t) &= \arg\min_{u_1 \in U_1} \Big\{ L_1(x(t), u_1^*(t), u_2^*(t), t) + p_1 \cdot f(x(t), u_1^*(t), u_2^*(t), t) \Big\}, \\
\dot{p_2}(t) &= -p_2(t)\frac{\partial f}{\partial x}(x(t), u_1^*(t), u_2^*(t), t) - \frac{\partial L_2}{\partial x}(x(t), u_1^*(t), u_2^*(t), t), \\
u_2^*(t) &= \arg\min_{u_2 \in U_2} \Big\{ L_2(x(t), u_1^*(t), u_2^*(t), t) + p_2 \cdot f(x(t), u_1^*(t), u_2^*(t), t) \Big\}, \\
x(0) &= x_0, \\
p_1(T) &= \nabla \psi_1(x(T)), \\
p_2(T) &= \nabla \psi_2(x(T)),
\end{aligned}} \tag{2.8}
$$

where a new co-state equation and control input appear to accomodate for the second cost functional.

2.1.3 Closed-loop Strategies

In many cases, the players of a DG have available the current state of the game, and use that information for their advantage. When the agents are allowed to use the information of the current state of the game, and react according to that information, their strategies are refered as *closed-loop* or *feedback strategies*. Hence, for the two-player DG defined by (2.3) and (2.5), the goal is to find a pair of control inputs u_1^* and u_2^* that simultaneously minimize both performance criteria, by using the current state of the game.

As for the open-loop case, it is convenient to first analize the equivalent OCP, in order to achieve a better understanding of the solution. Traditionally, if one wanted to solve a problem of this nature using the current state as given information, the solution would imply having a control law. It is well known that this control law can be obtained by means of a tool called *dynamic programming* [Bre11]. Conceptually, dynamic programming is used to solve discrete time problems; nonetheless, it can be used to solve continuous time problems, such as an OCP. By extending Bellman's principle of optimality [BD62] to continuous time, it is possible to determine that the optimal cost associated to (2.1) and (2.2) can be determined by solving the following partial differential equation:

$$\boxed{\begin{aligned} &\frac{\partial J^*}{\partial t}(x(t),u(t),t) + \min_u \left\{ g(x(t),u(t),t) + \nabla_x J^T f(x(t),u(t),t) \right\} = 0, \\ &\hspace{6cm} J^*(x(T),u(T),T) = G(x(T),t), \end{aligned}} \tag{2.9}$$

known as the *Hamilton-Jacobi-Bellman* (HJB) equation, where $J^* \in \mathbb{R}$ is the optimal cost. From the second term in the HJB equation, it can be to notice that there is a relationship between the optimal cost and the optimal control law, hence, by determining the optimal cost, the optimal control law can be obtained.

As it has been shown before, in a DG problem there is not a single cost functional to be optimized; there are two or more objectives that must be optimized simultaneously by the players in the game, thinking about the actions of others. This implies that in a DG there are several partial differential equations, one for each player, the determine the optimal cost for each functional. Since all cost functionals depend on the state of the game, which in turn depends on the actions of the agents, this partial differential equations are coupled. Thus, in order to solve the game described by (2.3) and (2.5), it is necessary to solve the following set of equations:

$$\boxed{\begin{aligned} \frac{\partial J_1^*}{\partial t}(x(t), u_1(t), u_2(t), t) + \min_{u_1} \Big\{ L_1(x(t), u_1(t), u_2(t), t) + \\ \nabla_x J_1^T f(x(t), u_1(t), u_2(t), t) \Big\} = 0, \\ \frac{\partial J_2^*}{\partial t}(x(t), u_1(t), u_2(t), t) + \min_{u_2} \Big\{ L_2(x(t), u_1(t), u_2(t), t) + \\ \nabla_x J_2^T f(x(t), u_1(t), u_2(t), t) \Big\} = 0, \\ J_1^*(x(T), u_1(T), T) = \psi_1(x(T), T), \\ J_2^*(x(T), u_2(T), T) = \psi_2(x(T), T), \end{aligned}} \tag{2.10}$$

where the equations are coupled by the state of the game, as well as the L_i functions referenced in (2.5).

2.2 Mean Field Games

MFGs are a branch of game theory that models dynamic decision making in scenarios where the number of players is large. The main idea behind MFG is to avoid modeling all the interactions among the players on the game, and only model the interaction between the agents and the *mass* of others. This mass is related with the distribution of the state of the players in the space, and can be modeled using statistichal mechanics tools [LL07, HCM03].

As it has been stated before, in order to solve an N player DG in which the agents are using feeback strategies, it is required to solve a set coupled HJB PDEs. Although this is possible when N is not large, it becomes challenging as N grows, i.e., when the number of players in the game grows. For that matter, a novel tool is required to solve this kind of games, so that the optimal strategies for large numbers of agents can be determined. In order to study this large-scale games, it is proposed to analyze the behavior of a single agent and the bulk properties of others. In that sense, a single agent is not concerned about his single interaction with every other player, but rather to the aggregated effect of many individuals. Thus, large populations can be simplified into a single variable that collects all the information to describe the mass of agents.

In order to study a set of players as a single mass, it is convenient to study their statistical properties. This simply means that in a MFG, agents are not concerned about the state x_i of every other player, but rather to how these states are distributed in the state space. However,

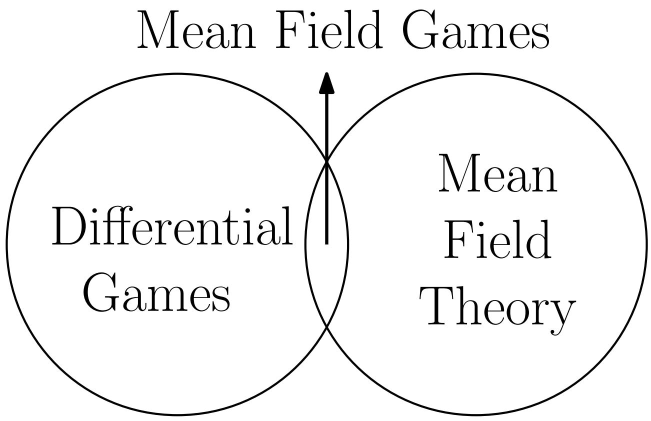


Figure 2.2: Relationship between DG and mean field theory.

this only makes sense if the number of agents in the game is large enough so that the action of a single individual does not affect the overall evolution of the system, i.e., a single agent is not able to lead the game by itself. This behavior has been studied by physicists in the framework of statistical mechanics, and thus, it is a suitable idea to borrow the mathematical tools from their framework in order to model MFGs. Figure 2.2 shows the relationship that exists between DGs and statistical mechanics (by using mean field theory).

Consider a set of N agents playing a DG in which N is sufficiently large so that the actions of a single agent do not affect the overall state of the game. Also, consider that all agents belong to the same population, i.e., all agents can be considered as subsystems with *similar* state equations. By saying that the agents have similar state equation, it is understood that if one look at the state equation of a representative agent of the population, i.e., a fictional agent that shares the similar characteristics with others in the population, his state equation would be as follows:

$$d\mathbf{x} = f(\mathbf{x}(t), \mathbf{u}(t), t)\, dt + \sigma\, d\mathbf{B}(t), \tag{2.11}$$

where $f(\cdot, \cdot, \cdot)$ is a function that generalizes the dynamics of all agents, and $\sigma\, d\mathbf{B}$ is a stochastic component that indicates that agents are not exactly alike. Notice that $\sigma\, d\mathbf{B}$ can be considered as a parameter that models the differences in the evolution of the states of the players. It is important to point out that if all agents have the same state equation, (2.11) would be exactly the function that determines the evolution of any agent.

It has been stated that in a MFG agents are only concerned about the distribution of the states of others. Given that the states of the players are evolving in time, the distribution of their states also evolves. Hence, this distribution of the states can be written as a variable $m(\mathbf{x}, t)$ that determines how the entries of the state vector of the game are distributed at each time instant. Given that the representative agent of the population gives the best characterization of the agents of the system, it is possible to say that its state variable is a random variable that is distributed as m. In statistical mechanics, it is well-known that the evolution of the probability distribution of a random process given by an controlled state equation is given by a PDE known as the *Fokker-Planck-Kolmogorov* (FPK) equation. Hence, the evolution of the $m(\mathbf{x}, t)$ is given by

$$\frac{\partial m}{\partial t}(\mathbf{x}, t) + \nabla \cdot (m(\mathbf{x}, t)\, f(\mathbf{x}, \mathbf{u}, t)) = \frac{\sigma^2}{2} \Delta m(\mathbf{x}, t), \tag{2.12}$$

where $\Delta = \sum_{i=1}^{n} \frac{\partial^2}{\partial x_i^2}$ is the Laplacian operator. Notice that in Equation (2.12) possesses the function f that determines the evolution of the representative agent of the population. Equation (2.12) gives the information about how the state vector is distributed at all time instants, and thus, agents can use this information to make their optimal decisions. It is also important to point out that 2.12 does not give information about the current state of each player of the game, as this information is not required for the computation of optimal strategies (as will be seen next).

The considered representative agent of the population behaves as every other agent in the game. Hence, he is trying to optimize a given performance criteria, but now the information about the mass of others is known, i.e., the function $m(\mathbf{x}, t)$. The representative agent of the population is minimizing the following performance:

$$J(\mathbf{x}, \mathbf{u}, m, t) = E\left[\int_0^T g(\mathbf{x}, \mathbf{u}, m, t)\, dt + G(\mathbf{x}, m, T) \right], \tag{2.13}$$

where g and G only depend on the current state of the representative agent and the distribution of others. Notice that since the representative agent is only a mere generalization of every agent, solving the OCP for this agent derives in the best control law for every agent.

The minimization of the cost functional (2.13) subject to the state equation (2.11) can be written as a HJB as in the case of the DGs. The HJB equation associated with the representative agent is as follows:

$$\frac{\partial J}{\partial t} + \mathcal{H}(\mathbf{x}, \mathbf{u}, \nabla_x J, m, t) + \frac{\sigma^2}{2} \Delta J = 0. \tag{2.14}$$

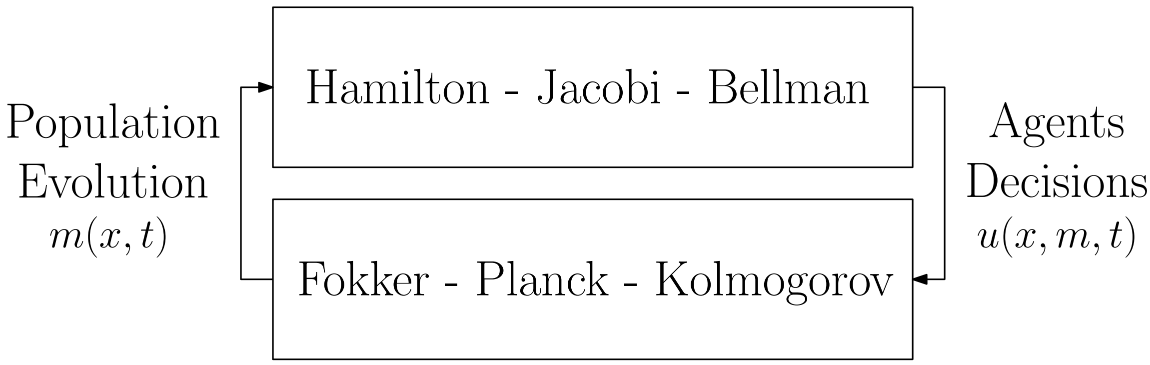


Figure 2.3: MFG system of PDEs.

Since this HJB equation uses the information from the distribution of the system, it is coupuled with the FPK presented previously. This means that in order to determine the best strategy for the representative agent, and in term for every agent, it is required to solve a set of two PDEs. This set of PDEs is know as the HJB-FPK system of PDEs, and is a canonical set for the MFGs. The relationship that exists between the distribution of the players and their actions is summarized in Figure 2.3.

The canonical system for the MFGs is sumarized next:

$$\boxed{\begin{aligned} \frac{\partial J}{\partial t} + \mathcal{H}(\mathbf{x}, \mathbf{u}, \nabla_x J, m, t) + \frac{\sigma^2}{2}\Delta J = 0, \\ \frac{\partial m}{\partial t}(\mathbf{x}, t) + \nabla \cdot (m(\mathbf{x}, t)\, f(\mathbf{x}, \mathbf{u}^*, t)) = \frac{\sigma^2}{2}\Delta m(\mathbf{x}, t), \\ J(\mathbf{x}, m, T) = G(\mathbf{x}, m, T), \\ m(\mathbf{x}, 0) = m_0. \end{aligned}} \tag{2.15}$$

The system (2.15) characterizes the equilibrium of the game; if one is able to find a control law u^* and a distribution m^* that simultaneously solve the system, the Nash equilibrium can be found. This fact mean that, in order to find the best strategies for a large scale DG in which the dynamics of the agents are similar, it is only required to solve two PDEs. Notice that this problem can be seen as a traditional OCP with an extra constraint: the FPK equation.

2.3 Summary

This chapter has presented the basics of game theory required to understant the content of this thesis. The DGs and the MFGs have been briefly enunciated, as well as how they can be solved. The notion behind the Nash equilibrium of a dynamic game has been stated. The main difference between closed-loop strategies and open-loop strategies has been also stated. Likewise, the main theoretical differences between MFGs and DGs have been enunciated.

CHAPTER 3

DIFFERENTIAL GAME APPROACH

This chapter is entirely based on:

A. RAMIREZ-JAIME, N. QUIJANO, C. OCAMPO-MARTINEZ. *A Differential Game Approach to Urban Drainage System Control.* American Control Conference*, 2016. (*Accepted*)*

3.1 Introduction

This chapter proposes an introductory scheme in which only a DG is considered. This allows to understand the main concepts behind dynamic games and CSN in a real case study, but also, allows to point out the main dissadvantages of a full DG approach. This leads to the necessity of use of techniques such as the MFG.

3.2 Problem Statement

3.2.1 Urban drainage system model

Water running through the pipes of a UDS can be modeled by using the so-called Saint-Venant equations (SVEs), which use mass and momentum conservation principles, in order to describe the phenomena occurring inside the pipes [Cho59]. These equations describe in a quite high level of detail how water flows. However, that level of detail is usually not required for control design. For that reason, a control-oriented model based on the *virtual-reservoirs* model is used. In this approach, the UDS is divided into a set of interconnected real and virtual tanks (VT).

According to [OM10], a VT is a storage element that represents the total volume of wastewater inside the pipes associated to a determined portion of the network. The volume is calculated via a mass balance equation, so that the state equation of the i-th tank is given by

$$\dot{v}_i(t) = q_i^{in}(t) - q_i^{out}(t), \quad \forall i \in \{1, \ldots, N+1\}, \tag{3.1}$$

where v_i is the total volume of water of the reservoir, q_i^{in} and q_i^{out} are the total inflow and outflow coming in and out of the i-th reservoir, and $N+1$ is the total number of reservoirs. For this thesis, it is assumed that the reservoirs are linear, and therefore the outflow of every tank is proportional to the volume stored in it, i.e., $q_i^{out}(t) = k_i v_i(t)$, where k_i is a *volume/flow* conversion (VFC) constant.

Finally, it is assumed that there is a retention gate located at the output of every VT. This implies that the manipulated inputs of the system are the outflows of VTs, which can be adjusted by opening or closing the gates. Since the outflow is proportional to the volume of the reservoir, the following constraint must hold:

$$0 \leq u_i(t) \leq k_i v_i(t), \quad \forall t \in \mathbb{R}_+, \tag{3.2}$$

where u_i is the manipulated outflow of the i-th reservoir.

The tree-shape topology of the UDS can be simplified by using the VT model by saying that any portion of a given UDS can be seen as a collection of tanks, whose outflows converge into a common reservoir, until the outlet reservoir is reached. For instance, Figure 3.1 shows a typical tree topology with 21 VTs of a UDS after the simplification is used. Notice how the whole network eventually converges to the drain reservoir v_{drain}, and every VT (least the drain VT) has a retention gate located at the output that regulates the outflow. Hence, (3.1) can be written as

$$\dot{v}_i(t) = -u_i(t) + \sum_{j \in \mathcal{S}_i} u_j(t) + d_i(t), \tag{3.3}$$

where $\mathcal{S}_i$ is the set of tanks whose outflows go directly to the i-th tank, d_i is the total inflow from rainfall entering the i-th reservoir, and knowing that constraint (3.2) must be satisfied at all times. It is important to notice that d_i acts as a disturbance that alters the state of the i-th tank. Since the last tank of the network does not have a retention gate at the output, its state equation is written as

$$\dot{v}_{N+1}(t) = -k_{N+1} v_{N+1}(t) + \sum_{j \in \mathcal{S}_{N+1}} u_j(t) + d_{N+1}(t), \tag{3.4}$$

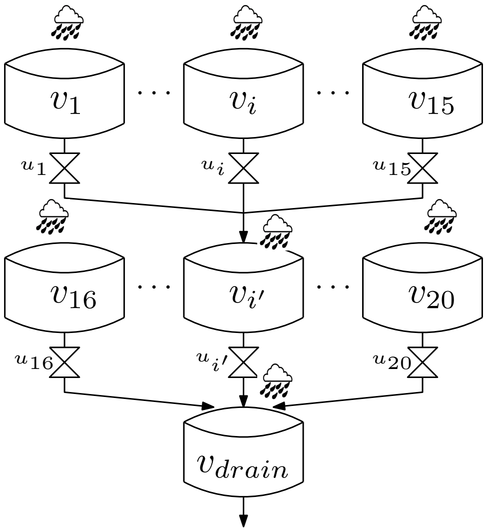


Figure 3.1: Typical tree topology of a UDS after the simplification based on VTs is applied.

where $\mathcal{S}_{N+1}$ is the set of tanks whose outflows go directly into the drain tank. Equation (3.4) states that the drain tank cannot control its outflow.

3.2.2 Information graph

A directed graph can be used to represent the interactions among the tanks of a UDS [JDJOM$^+$14]. This representation gives useful insights on how water moves throughout the pipes, but it is also useful in distributed control design because it can describe communication structures among local controllers. On an usual representation, a vertex of the graph corresponds to a reservoir of the network, and edges represent the flow of water among the reservoirs of the network. The edges of an usual UDS graph representation, e.g., [JDJOM$^+$14], are oriented in the direction of gravity, i.e., from an upstream tank to a downstream one. For this thesis however, the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N, v_{drain}\}$ and $\mathcal{E} = \{(v_i, v_j) \in \mathcal{V} \times \mathcal{V} \mid v_j \in \mathcal{S}_i\}$, represents the UDS. Notice that the only difference with the standard representation is the orientation of the edges. Define $\mathcal{N}_i = \{v_j \in \mathcal{V} \mid (v_j, v_i) \in \mathcal{E}\}$ as the neighborhood of the i-th reservoir, i.e., all the reservoirs to where the outflow of the i-th tank is going to. This neighborhood describes the information that is available to each agent located at each vertex of the graph, i.e., the local controller responsible for the manipulation of the outflow of single reservoir. The reason for the selection of this graph architecture is that, in order to determine its best action, every local controller should know all the states that its local actions are altering, i.e., all the tanks of its neighborhood.

3.3 Differential Game Controller

The proposed scheme uses a distributed consensus algorithm based on a DG, in the same spirit as in [Gu08], which seeks agreement on the normalized volume of the tanks. This allows for the retention of the proper amount of water upstream, and guarantees an even use of the reservoirs.

According to [BO95], the so-called dynamic game theory studies multi-player decision making in situations where not only the actions that players (also known as agents) make are important, but also the order in which they are made. This means that the game is going to evolve over time following the actions that have been made by the agents. Analogously, DG theory studies multiplayer dynamic decision making in situations where the evolution of the game can be described by a set of first-order differential equations. Then, it can be said that DG theory studies the optimal control of dynamical systems that have several independent manipulated inputs. This framework allows for the design of distributed optimal control strategies for dynamical systems with several inputs (both manipulated and non-manipulated) [BB08].

For the proposed DG, an agent is a local controller that is responsible for the control of one retention gate. This agent can be seen as the dynamical system composed of the i-th tank and the i-th retention gate. This agent has available the volume stored in the reservoirs of its neighborhood, e.g., agent i-th has available the volumes of the tanks in $\mathcal{N}_i$. The goal of this agent is to change the outflow of a reservoir in order to achieve an even normalized volume on the tanks of its neighborhood. Since the decisions that a single agent makes have an inpact on the game, and thus on other agents, the standard optimal control tools cannot be applied directly, and DG theory must be used instead.

According to [BO95], to properly define a DG, it is necessary to define a state equation that describes the evolution of the game, and a set of cost functionals to be optimized by the players. State equations (3.3) and (3.4) can be written in matrix form as

$$\dot{\mathbf{v}}(t) = \mathbf{A}\mathbf{v}(t) + \sum_{i=1}^{N} \mathbf{B}_i u_i, \tag{3.5}$$

where $\mathbf{v} = [v_1, v_2, \ldots, v_N, v_{drain}]^\top \in \mathbb{R}^{N+1}$, and $\mathbf{A}$ and $\mathbf{B}_i$ are matrices of proper dimensions.

Define $\mathbf{v}$ as the state of the game. On this formulation, each agent computes one u_i and seeks the minimization of

$$J_i = \int_0^T \Big\{ \sum_{j \in \mathcal{N}_i} w_{ij} (\bar{v}_i(t) - \bar{v}_j(t))^2 + r_i u_i^2(t) \Big\} \, dt, \tag{3.6}$$

where $w_{ij} \geq 0$ and $r_i > 0$ are weightening factors, and T is the duration of the game. Therefore, (3.6) can be written in compact form as

$$J_i = \int_0^T \Big\{ \bar{\mathbf{v}}^\top(t) \mathbf{Q}_i \bar{\mathbf{v}}(t) + r_i u_i^2(t) \Big\} \, dt, \tag{3.7}$$

where $\mathbf{Q}_i \geq 0$. Cost function (3.7) states that the i-th agent tries to seek an agreement on the normalized states of its neighborhood, while using a minimum amount of energy in the process. In this thesis, the following assumption is to be made in order to have a proper definition of a DG.

The simultaneous minimization of the functionals (3.7) subject to the state equation (3.5) describes a linear-quadratic (LQ) differential game (DG) [BO95].

3.3.1 Nash equilibrium

The solution to the previous DG requires the simultaneous minimization of cost functionals that are, in general, not the same. Hence, the notion of optimality is not as clear as in a standard optimal control theory, because there is no single criteria for what an optimum is. In traditional game theory, the notion of optimality is augmented into the notion of equilibrium, and thus allowing the search of a solution to the previous problem [BO95]. There are several different types of equilibria that can be found in a DG. For instance, if one of the agents announces its strategy before hand and every other agent reacts to that doing, the optimal behaviour of the agents is known as a *Stackelberg* equilibrium [BO95]. For this work however, only the so-called *Nash* equilibrium (NE) is studied. An NE is a set of strategies where no agent can improve its payoff by changing its strategy while others keep theirs fixed [Nas50]. According to [Eng05], a set of actions $(u_1^*, u_2^*, \ldots, u_N^*)$ is an NE for an N-player game, where each player is trying to minimize J_i, if for all $(u_1, u_2, \dots, u_N)$ the following inequalities hold:

$$\begin{aligned} &J_i(u_1^*, u_2^*, \dots, u_i^*, \dots, u_N^*) \\ &\leq J_i(u_1^*, u_2^*, \dots, u_i, \dots, u_N^*), \quad \forall i \in \{1, 2, \dots, N\}. \end{aligned} \tag{3.8}$$

Then, it can be said that an NE of the game is a set of strategies where u_i^* is the best response for the i-th agent, regardless of what any other agent is doing. Since the proposed controller derives into an LQ DG, it makes sense to study the NE within that framework.

The study of DGs requires the knowledge of the information pattern associated to each player. The information pattern is the information that a player is allowed to have throughout the duration of the game. Two information patterns are usually analyzed on DG theory: open-loop information patterns, and feedback information patterns [BO95]. The difference between these patterns is whether or not an agent is allowed to have the current state of the game. It is important to point out that, although on the open-loop information pattern agents are not able to measure the state vector of the game, they do know what the initial condition is. In this thesis, only open-loop information patterns and their associated NE are studied, due to the simplicity of its analysis.

The following theorems have been adapted from [Eng05, Th. 7.1], [Eng05, Th. 7.2], and [Eng05, Col 7.3.], for this particular application, and defining $\mathbf{S}_i = \mathbf{B}_i r_i^{-1} \mathbf{B}_i^\top$.

Theorem 3.1. *The N-player DG described by (3.5) and (3.7) has an unique open-loop NE for every intial state $\mathbf{v}(0)$ if and only if $\det(\mathbf{H}(T)) \neq 0$, where*

$$\mathbf{H}(T) = [\mathbb{I}_{N+1} \; 0_{N+1} \; \dots \; 0_{N+1}] e^{-\mathbf{M}T} \begin{bmatrix} \mathbb{I}_{N+1} \\ 0_{N+1} \\ \vdots \\ 0_{N+1} \end{bmatrix},$$

and

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{S}_1 & \mathbf{S}_2 & \dots & \mathbf{S}_N \\ -\mathbf{Q}_1 & -\mathbf{A}^\top & 0_{N+1} & \dots & 0_{N+1} \\ -\mathbf{Q}_2 & 0_{N+1} & -\mathbf{A}^\top & \dots & 0_{N+1} \\ \vdots & & & \ddots & \\ -\mathbf{Q}_N & 0_{N+1} & \dots & 0_{N+1} & -\mathbf{A}^\top \end{bmatrix}.$$

Theorem 3.2. *If $\mathbf{H}(T)$ is invertible, then the set of coupled Riccati equations*

$$\dot{\mathbf{P}}_i = -\mathbf{A}^\top \mathbf{P}_i - \mathbf{P}_i \mathbf{A} - \mathbf{Q}_i + \sum_{j=1}^{N} \mathbf{P}_i \mathbf{S}_j \mathbf{P}_j, \quad \mathbf{P}(T) = \mathit{0}_{N+1},$$

has a unique solution in $[0, T]$ and the set of strategies

$$u_i^*(t) = -r_i^{-1} \mathbf{B}_i^\top \mathbf{P}_i(t) \mathbf{\Phi}(t, 0) \mathbf{v}(0),$$

characterizes the NE of the game, where

$$\dot{\mathbf{\Phi}}(t, 0) = (\mathbf{A} - \sum_{i=1}^{N} \mathbf{S}_i \mathbf{P}_i) \mathbf{\Phi}(t, 0), \quad \mathbf{\Phi}(t, t) = \mathbb{I}_{N+1},$$

is the state transition matrix of the closed-loop system.

Theorems 1 and 2 determine the existence and uniqueness of the solution to the simultaneous minimization of the cost functionals (3.7) subject to the state equation (3.5) in terms of a NE, for every initial state of the system. This means that, the solution of the game for the UDS problem depends heavily on parameters that can be selected, e.g., T and r_i. Thus, they can be chosen so that the solution always exists for every $\mathbf{v}(0)$. This is particularly useful because the proposed methodology uses an open-loop information pattern to compute the optimal strategies, and a receding horizon approach is needed to give feedback to the solutions [Gu08]. Moreover, since Theorem 3.2 presents a way to compute the open-loop strategies, they can be calculated easily for recursive approaches, such as the receding horizon approach.

As noted by [Gu08], it might seem that, in order to compute the optimal strategies u_i^*, every agent requires the whole state vector $\mathbf{v}(0)$. However, since matrix $\mathbf{Q}_i$ only has non-zero entries at the positions of $\mathcal{N}_i$ and i itself, and given that the matrix $\mathbf{A}$ has a diagonal structure, the solutions $\mathbf{P}_i$ of the coupled Riccati equations only have non-zero elements at the positions of the i-th agent and its neighborhood. Therefore, u_i^* is in fact a distributed control law.

3.3.2 Receding horizon DG

In general, open-loop control strategies are not able to react against those disturbances that may alter the state of the system. Hence, the open-loop strategies developed previously are not able to work sussesfully for the addressed problem, since the control of UDS depends heavly on the disturbances (rainfall) that alter the system. However, a receding horizon scheme [Gu08] can be used to take care of that problem and add feedback to the overall law.

The set of strategies u_i^* are determined at $t = 0$ for the time interval $[0, T]$, which means that the system only knows how it should behave during that period. However, that set of strategies can be recomputed at $t = t_1$ for time interval $[t_1, T + t_1]$, so that the system knows how it should behave at a different time interval. To recompute the strategy, the system has to measure the initial state (which is now $\mathbf{v}(t_1)$) at a new time, hence a feedback appears. This process is repeated for $\{t_2, t_3, \ldots\}$ until any desired final time is reached. This scheme is known as a receding horizon scheme and allows to have feedback on DG with open-loop information patters.

The algorithm used in this thesis is shown in Algorithm 3.1, where t_f denotes the final time of the simulation scenario.

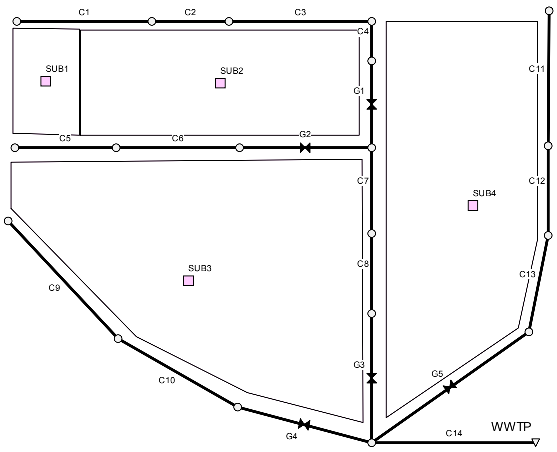
Algorithm 3.1 Algorithm used for the receding horizon DG.

While $t \neq t_f$:
{ Measure current state vector $\mathbf{v}(t_i)$
Solve the DG $\forall t \in [t_i, T + t_i]$
Compute the optimal stretegies u_i^* for $\mathbf{v}(t_i)$
Apply u_i^* during $[t_i, t_{i+1}]$
$t_i = t_{i+1}$ }

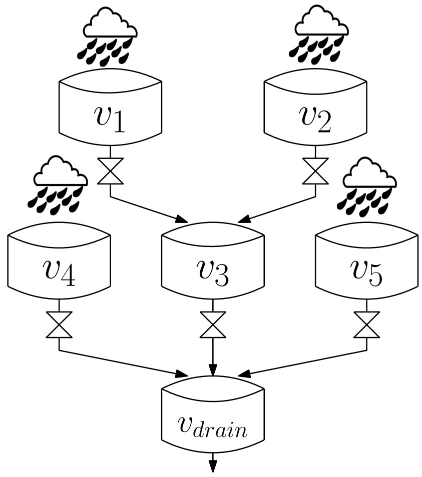
Algorithm 3.1 is based upon discrete time increments, while the analysis for the computation of the optimal strategies is done on continuous time. This implies that during two discrete time instants, the system is applying a continuous time function that evolves over time, and thus, the system is using an open-loop law between two consecutive discrete times. This is different from traditional receding horizon approaches [Mac02], because the control input is able to change in between a given time interval. However, during the computational implementation of the controller, it is required to have a time discretization on the system, and thus, on the optimal strategies. If the selected sampling time for the implementation of the controller is short enough, it can be assumed that the strategies do not change during a time interval. Hence, for the implementation of the system, constant functions are applied during $[t_i, t_{i+1}]$.

3.4 Case Study

The proposed controller is tested with the network shown in Figure 3.2a. This UDS is composed of 4 sub-catchments that drain into a tree-like network that, in turn, converges into a common outlet node. This network gives a convenient representation of how a full-size UDS would look like, because of its strong convergence topology. Moreover, this network allows to study one of the most common problem associated with UDSs, which is the uneven use of the pipes of the system, which leads into poor wastewater management, and in most cases, flooding. Hence, this network is a suitable testbed for determining the performance of controllers of UDSs.



(a) Tree-shape network.



(b) Simplified network.

Figure 3.2: Case study. a) Proposed network for the testing of the control strategy; and b) Equivalent model of the UDS after the VT simplification is used.

The system is simplified into a set of 6 interconnected VTs using the virtual-reservoir model, where a single tank corresponds to all the pipes in between retention gates, or a retention gate and a inlet or outlet node. The simplified network is shown in Figure 3.2b, where the sub-catchments drain directly into Tanks 1, 2, 4 and 5. Since an information graph is necessary to describe the information available to each agent, and following the definition of Section 3.2.2, Figure 3.3 shows the associated graph of the proposed case study.

This model requires the calibration of two parameters: the VFC and v^{max} for each reservoir.

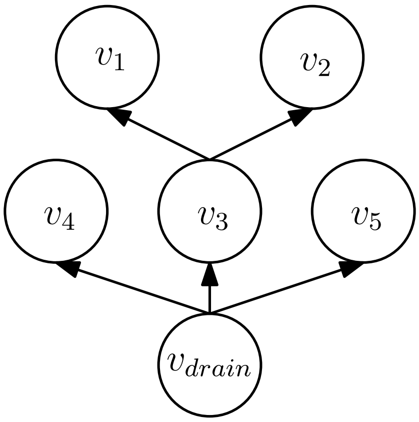


Figure 3.3: Information graph that determines the information available to each agent in the case study.

Table 3.1 presents the parameters asociated with the simplified network. They have been obtained using simulated data gathered from MatSWMM [RBRJBG+15], which is a co-simulation tool for UDS with *Matlab* and *EPA-SWMM*. Due to the fact that the sub-catchments have differ-

Table 3.1: Parameters of the simplified model

Reservoir	**VFC** $\times 10^{-3}$ [s^{-1}]	v_i^{max} $\times 10^4$ [m^3]
v_1	0.4	1.9543
v_2	0.8	0.2933
v_3	3.3	0.3578
v_4	0.5	0.2762
v_5	0.8	0.2976
v_{drain}	1.8	0.3032

ent geographical locations, they recieve different amounts of rainfall, and thus the total inflow entering each reservoir is different. Figure 3.4 shows the rain scenario proposed for this application where, d_i represents the total inflow entering the i-th reservoir.

The key idea is to test how the network reacts with the proposed rain scenario, which in fact generates overflow downstream of the UDS. Then, it is going to be determined whether or not the proposed methology is able to manage the tanks eficiently so that the flooding is minimized and the water resourse evently distributed.

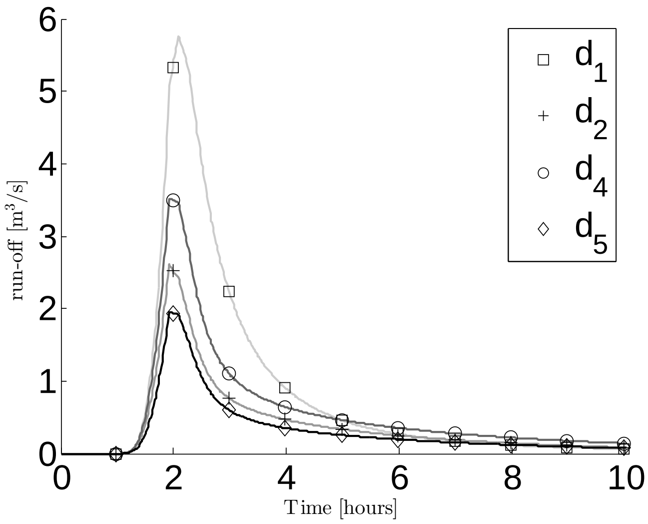


Figure 3.4: Proposed precipitation event for the case study. d_i corresponds to the total inflow entering the i-th reservoir of the system.

3.5 Results and Discussion

In order to determine the actual performance of the proposed scheme, the system is tested for three different cases: i) the UDS with no controller on the loop; ii) the UDS being controlled via a centralized model predictive control (MPC); and iii) the UDS being controlled via a distributed control strategy based on a DG. The performance of the controllers is analyzed upon how effenciently the reservoirs are being used, and what the total flooding is.

The centralized MPC used for comparison has the following cost function:

$$
\begin{aligned}
J_{MPC} = \sum_{k=0}^{H_p-1} & w_{13}(\bar{v}_1(k) - \bar{v}_3(k))^2 + w_{23}(\bar{v}_2(k) - \bar{v}_3(k))^2 \\
& + w_{3d}(\bar{v}_3(k) - \bar{v}_d(k))^2 + w_{4d}(\bar{v}_4(k) - \bar{v}_d(k))^2 \\
& + w_{5d}(\bar{v}_5(k) - \bar{v}_d(k))^2 + r_1 u_1(k)^2 + r_2 u_2(k)^2 \\
& + r_3 u_3(k)^2 + r_4 u_4(k)^2 + r_5 u_5(k)^2,
\end{aligned}
$$

which can be written in compact form as

$$
J_{MPC} = \sum_{k=0}^{H_p-1} ||\bar{\mathbf{v}}(k)||^2_{\mathcal{L}} + ||\mathbf{u}(k)||^2_{\mathbf{R}}, \tag{3.9}
$$

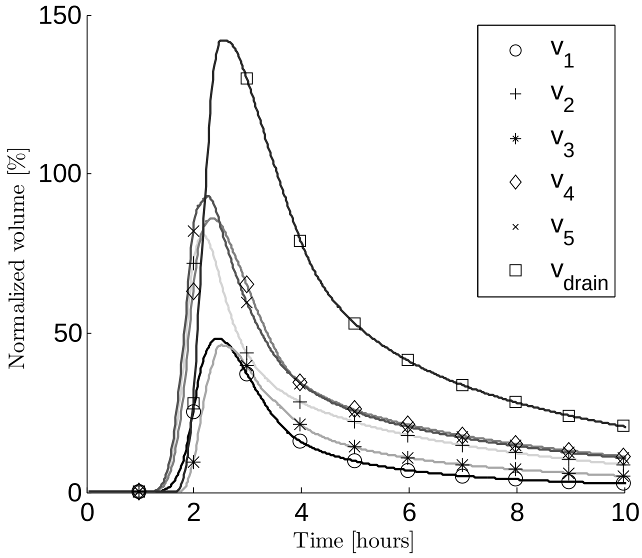
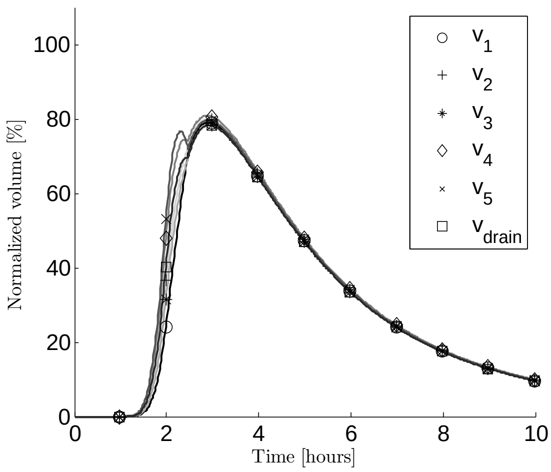


Figure 3.5: Open-loop response of the system during the proposed precipitation event. It shows the evolution of the normalized volumes for a 10-hour window.

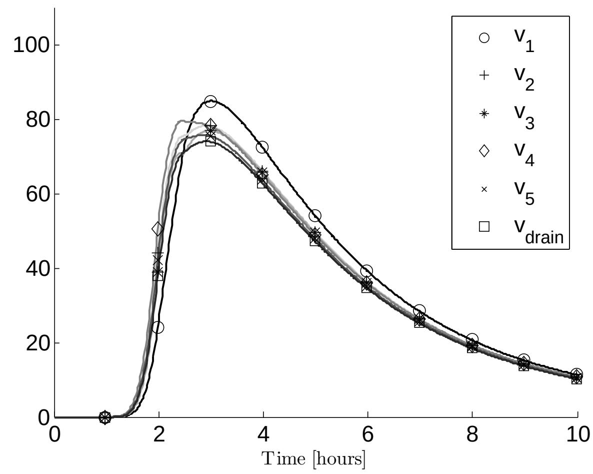
where $\mathbf{u} = [u_1, u_2, \ldots, u_N]^\top$, $\mathcal{L}$ is the Laplacian matrix of the graph with edge weights w_{ij}, $\mathbf{R} > 0$ is a diagonal matrix with all r_i on the diagonal, H_p is the prediction horizon, and $k \in \mathbb{Z}_+$ denotes the discrete time. It is important to point out that H_p has been selected so that the prediction window matches the duration a the DG, i.e., T. The reason for the selection of this cost function is to have a consensus-like algorithm in the centralized MPC, so that both controllers, i.e., MPC and DG-based, have the same overall goal.

As for case i), Figure 3.5 shows the open-loop response of the system, i.e., when there are no controllers manipulating the retention gates, for a 10 hours window. It is shown that there is not a proper management of the reservoirs, since they are not evently used. For instance, some reservoirs, such as 1 and 3, remain underused, while the drain tank presents an overuse of about 50% of its maximum capacity. This is due to the fact that upstream reservoirs only recieve water from rainfall, whereas the downstream one recieves wastewater from many different tanks which, in term, recieve from rainfall. Therefore, upstream tanks can retain water in order to release some of the burden existing downstream, which leads to a better usage of the existing network.

Cases ii) and iii) evaluate the performance of the system when two different controllers are



(a) MPC controller on the loop.



(b) DG-based controller on the loop.

Figure 3.6: Closed loop response of the system when 2 controllers are applied. a) Centralized MPC controller; and b) Distributed DG controller. The normalized volumes of the tanks for a 10 hours window is shown.

used. Both controllers have the same objective: to seek and agreement on the normalized volume of the tanks. The main difference between both schemes is that the MPC uses centralized information in order to calculate the optimal outflows, whereas the DG only uses local information to achieve its goal. Figure 3.6 shows the evolution of the normalized volumes for a 10 hours window, when the two controllers, i.e., MPC and DG-based, are applied. Both controllers are able to completely remove the overflow found in the open-loop scenario. As for the MPC (Figure 3.6a), the normalized volumes become almost identical after a short time. This is a sign of proper management of the reservoir, which proves that the control strategy fulfills its goal. The controller based on the DG (Figure 3.6b) also completely removes the flooding from the network, and is able to manage the VTs so that their normalized volumes move close together.

The proposed scheme achieves a similar performance compared to a centralized controller in terms of flooding minimization and wastewater management. Nonetheless, the MPC requieres a lot of extra computational resources, which is a major problem in a large-scale system. For instance, the simulation of the system for a 10 hours time window takes 1 time unit to complete for the DG-based controller within the loop, whereas it takes 27 time units to complete with the MPC controller on the loop. These data have been obtained by using normalized values with respect to the fastest time, from the times collected using Matlab routines. The reason for this normalization is to have a simpler comparison between the measured times. Hence, a methodology such as the one proposed in this work thrives in large-scale problems where having optimization-based controllers is convenient.

3.6 Conclusions and into MFGs

Although the previous approach is quite useful for some applications, i.e., when an aggregated system is used, it can be challenging when the number of agents grow. This is due to the extra Riccati equations that must be solved, that cause the explicit solution of the DG to become untractable. For that matter, it is important to utilize tools that allow to solve DG that involve large numbers of agents, i.e., the MFGs.

CHAPTER 4

CONTROLLER DESIGN BASED ON DYNAMIC GAMES FOR CSN

This chapter collects the ideas that link the dynamic games shown in Chapter 2, with the real-time control of a combined sewer system (CSN). It shows some key concepts such as the definition of *agent* and the idea of his rationality. Moreover, it shows how this type of game is directly applied as a control strategy for CSN.

4.1 The Dynamic Game Definition

Typically, UDS have a strong convergence topology where many pipes end up into a common outlet node, until the drain node is reached. This causes most of the burden to be suffered downstream and quite little burden to be taken upstream of the network. This means that, usually, upstream pipes remain underused, so they could retain some water (by using retention gates) in order to minimize overflows downstream. The proposed scheme aims to solve that problem, by using a decentralized controller based on dynamic games, so that most of the pipes on the UDS are used efficiently, and the total overflow is minimized. Moreover, the proposed scheme seeks an integration with state-of-the-art techniques, e.g., MPC, to deliver more suitable results when other objectives are required, e.g., the minimization of combined sewer overflow (CSO).

4.1.1 Elements of the Dynamic Game

According to [BO95], in order to properly describe a dynamic game theory problem, it is required to state what an agent is and what his actuation mechanisms are for the selected application, in order to properly described how the strategy can be applied. Moreover, it is required to state how the different agents in the game are reasoning, i.e., what function they are trying to optimize, as well as how the overall game is evolving, i.e., the system dynamics. It is important to point out that these definitions are not unique and depend heavily on the proposed scheme.

The first required definition is the concept of an agent. Agents are local controllers that are able to manipulate the physical actuation mechanisms of the UDS. For instance, an agent might be a local controller capable of changing the inflow to a particular sewer pipe or storage unit. This means that in general, the agents are responsible for the selection of the appropiate control signals in order to guarantee a suitable operation of the system. This leads to the second required definition: the reasoning of the agents. As it has been state before, the reasoning of the agents are measuring functions that depend on the actions of the agents, and determine their performance in the game. For this thesis, the main goal of the agents is to distribute evenly the rainwater that enters into the network during a heavy rain event, so that no part of the system is prone to overflow due to the overuse of the capacity of the pipes and storage units. This reasoning can be captured by the minimization of the following cost function:

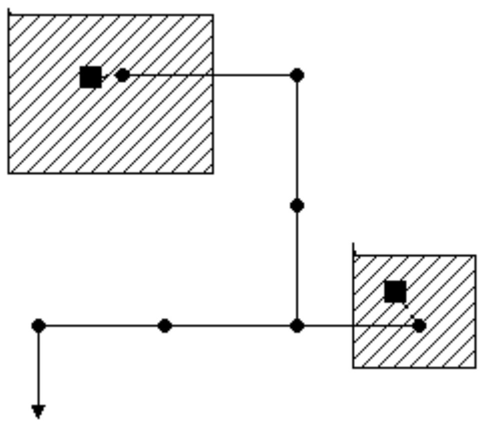
$$J_i = \int_0^T (x_i(t) - \phi(\mathbf{x}(t)))^2 + r u_i(t)^2 dt, \quad \forall i \in \{1, 2, \dots, N\}, \tag{4.1}$$

where x_i is the state of the system that the i-th agent is able to manipulate, $\phi(\mathbf{x})$ is a Lipschitz function of all the states of the system, u_i is the action that the i-th agent is taking, N is the total number of agents, T is a time horizon, and r is a weight parameter. It is important to point out that a single agent might be able to change multiple states of the system; if that is the case, the cost function would have a quadratic form of the states, instead of a simple substraction. The cost function (4.1) expresses the desire of each agent to change the states that his capable to manipulate, so that they become as close as possible to some function of the states system. Notice that having the function $\phi(\mathbf{x})$ is quite flexible in the sense that it allows to express the desire of a particular agent to modify the volume of a cluster of sewer pipes that are under his control, so that it becomes as equal as possible to the volume of some other cluster of pipes.

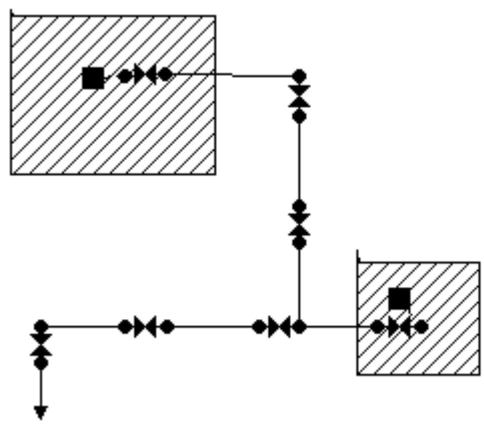
Finally, the third required definition is the dynamics of the game, which ultimately determine how the agents interact with each other and with the system. This definition is quite simple, given that the game dynamics derive from mass balance equations. This causes the state equations of the different sewer pipes to be linear with respect to the inflows and outflows to themself. The general evolution of the game is given by (this equation is going to be explined in more detail in the forthcoming sections):

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \sum_{i=1}^{N} \mathbf{B}_i u_i(t), \tag{4.2}$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices of proper dimensions. Notice that (4.2) is only able to give information about the mass balance of the UDS, since is not a hybrid state equation. It does not capture other phenomena such as the switching flow in weirs. Nonetheless, for the proposed control strategy, that is the only information that is going to be needed, because the strategies u_i are based on volumes exclusively, as it is going to be presented in the forthcoming sections.



(a) Simple network with no full controllability assumption.



(b) Simple network with the full controllability assumption.

Figure 4.1: After the model extention is performed to the system, each sewer pipe has a retention gate at its entrance. This causes all the inflows to sewer pipes to be controllable.

4.1.2 Model Extention

Typically, a UDS has a small amount of active elements, i.e., retention gates and redirection gates, which derives in a quite limited controllability of the flows that run through each sewer pipe. This may cause problems if a control strategy requires full controllability of all the states, and may lead to poor performances. For that matter, it is first assumed that there is a retention gate at the entrance of each sewer pipe in the network. This means that the inflow to each sewer pipe is completely controllable, and thus, there is going to be an agent of the game associated with that retention gate. This may seem like an unreasonable assumption, so it will be dropped latter on, as is only required for design purposes. Figure 4.1 illustrates how a simple network with six sewer pipes and two catchments is represented after the assumption has taken place. Notice that it would now be possible to control all flows running through the system.

Given this assumption, it is now known that the evolution on the game can be expressed in terms of the evolution of the volume of water inside each sewer pipe, because it is possible to relate one agent with each sewer pipe. Moreover, it is known that each agent is not only associated with a fictional retention gate, but with a volume of a pipe as well. Hence, it can be said that the dynamics of each agent of the game are given by

$$\dot{v}_i(t) = u_i - q_i^{out}, \tag{4.3}$$

where v_i is the state of the i-th agent, i.e., the volume stored in the i-th pipe, u_i is the action of the i-th agent, i.e., the controlled inflow to the i-th pipe, and q_i^{out} is the total outflow of the i-th pipe. It should be noted that there are some constraints in u_i, since the maximum inflow to a particular pipe cannot be greater than the total outflow from pipes whose outputs are directly connected to the i-th pipe, plus the rainwater entering to the network via i-th link. As it has been stated by [JD14], the outflow of a given sewer pipe can be expressed as a function of its inflow in a delayed time period, and thus, the state equation (4.3) can be simplified into an equation that only depends on the action of the agent itself.

It is interesting to point out that this model extention is a direct opossite of a model aggregation such as the virtual tank model [OM10], where many states of the system are associated to a single control variable, instead of adding a control variable for each state in the system. This relationship is useful, because it allows to use the proposed scheme for some aggregated representations of a UDS.

4.1.3 Differential Game Problem Statement

Consider a set of N agents playing a DG, where the evolution of each state of the system is given by (4.3), and each agent is trying to minimize

$$J_i = \int_0^T (v_i(t) - \gamma(\frac{1}{N}\sum_{j=1}^{N} v_j(t) + \eta))^2 + r u_i(t)^2 dt, \qquad \forall i \in \{1, 2, \dots, N\}, \tag{4.4}$$

where γ and η are known parammeters. The main problem is to find a set of actions $\{u_1^*, u_2^*, \dots, u_i^*, \dots, u_N^*\}$, i.e., inflows to sewer pipes, such that the inequalitites

$$J_i(u_1^*, u_2^*, \dots, u_i^*, \dots, u_N^*) \leq J_i(u_1^*, u_2^*, \dots, u_i, \dots, u_N^*), \tag{4.5}$$

are simultaneously satisfied $\forall i \in \{1, 2, \dots, N\}$. In other words, the problem is to find a Nash equilibrium (NE) for the proposed DG [Nas50, BO95, Eng05].

4.1.4 MFG Problem Statement

A typical UDS may have hundreds or even thousands of interconnected sewer pipes [JD14, BD04]. This causes the DG presented in the previous section to be large scale in nature, since there is a large amount of both states and control actions. For that matter, the solution to the game becomes untractable, and novel tools are required in order to find the NE. Thus, a MFG is proposed as a relief to large-scale problem in the DG. Following the basic MFG descriptions from [LL07] and [HCM03], it is assumed that the volume of water stored in the sewer pipes, i.e., the states of the game, behave as a random variable with probability distribution $m(v, t)$ at time t. This simply means that all the different volumes are condensed in a single variable. Finally, it is assumed that agents have available the probability distribution of the volumes, i.e., $m(v, t)$. It can be said that a MFG has a non-centralized information pattern, in which all agents interact indirectly using the distribution of all the system. This information pattern is illustrated in Figure 4.2.

For the MFG, the actions of the agents are based upon the information of the probability distribution of the volumes instead of the information from the actual volumes. This mean that a particular agent is more interested in the proportion of volumes that are in a particular level,

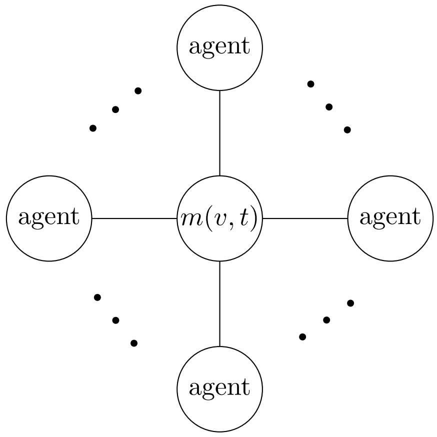


Figure 4.2: Non-centralized model of the MFG where agents have available the information of the distribution.

rather than the real volumes. Hence, the agents in the MFG game are trying to minimize

$$J_i^{MFG} = \int_0^T (v_i(t) - \gamma(\bar{m}(t) + \eta))^2 + ru_i^2 dt, \qquad \forall i \in \{1, 2, \ldots, N\}, \tag{4.6}$$

where $\bar{m}(t) = \int_{\mathbf{R}} vm(v,t)dv$ is the mean of the probability distribution of the volumes. Notice that (4.1) and (4.6) are identical if $N \to \infty$ in the differential game [LL07].

Thus, the main problem is as follows: consider a MFG with N agents, where the dynamics of a single agent are given by (4.3), and each agent is trying to minimize a cost functional such as (4.6). Then, the the problem is to find a set of actions $\{u_1^*, u_2^*, \ldots, u_i^*, \ldots, u_N^*\}$ such that the following inequalitites

$$J_i^{MFG}(u_1^*, u_2^*, \ldots, u_i^*, \ldots, u_N^*) \leq J_i^{MFG}(u_1^*, u_2^*, \ldots, u_i, \ldots, u_N^*) \tag{4.7}$$

are simultaneously satisfied $\forall i \in \{1, 2, \ldots, N\}$. In other words, the problem is to find a mean field NE of the game [HCM03, NCMH13].

In general, the MFG problem formulation is more suitable for typical UDS because they are large scale in nature. However, the DG problem formulation is useful when there is some kind of model aggregation in the representation and the number of states is *small*.

4.1.5 The Multi-population MFG definition

Up to now, the control problem has been focused around the minimization of overflows in the UDS. Although that objective is arguably the most important one, the controller of an UDS might have some other objectives, such as the transport of sewage to a WWTP. If it is desired to consider multiple objectives in the cost functional (4.6), the solution would become harder to compute, due to the extra complexity that the multiobjective optimization problem brings. A more convenient idea is to solve the multiobjective MFG by means of a multipopulation MFG (MP-MFG) [BMA14]. For this approach, each population in the game has a distinct goal, e.g., one population might focus on the even water distribution, while other might focus on the transport of the sewage to a WWTP. Even though it is possible to consider as many populations (and in term, as many objectives) as needed, for this thesis only two goals are considered: the minimization of overflows and the maximization of WWTP usage. Therefore, only two populations are required.

In a MP-MFG, the decisions of an agent from a particular population are based upon the shared information from its neighbor populations according to a known information graph. This scheme is quite flexible since a given population is not interacting directly with any agent from other populations, but rather with the populations as groups. This means that for this scheme, a single population does not care whether or not a MFG is occuring in the neighbor populations, because agents in the population only require specific information from the neighbors regardless of how it was determined. Thus, it is not required to have a MFG system on each node on the graph, which is convinient if one wants to use different control strategies combined. Figure 4.3 shows a possible configuration of 4 different MFG taking place simultaneously. In this configuration, each game shares information with its neighbors in order to guarantee a certain operation, as in the case of the UDS.

Consider a MFG as the one previously presented. In this approach, all agents are naturally seeking an even volume within all the pipes of the network. This water is stored, and in order to safetly evacuate it, it should be transported into a WWTP so that no recieving environment is damaged. The task of transporting the water to a WWTP is performed by an MPC based on the hyrbid linear delayed (HLD) model, whose objecive is the maximization of WWTP inflow and minimization of combined sewer overflow (CSO) [JD14]. Given that the MP-MFG approach does not require the interconnection of multiple MFG schemes, this MPC approach can be used combined with the MFG to minimize the overflows, so that both control objectives are achieved.

Nonetheless, a coupuling term is required to guarantee a suitable interaction between the

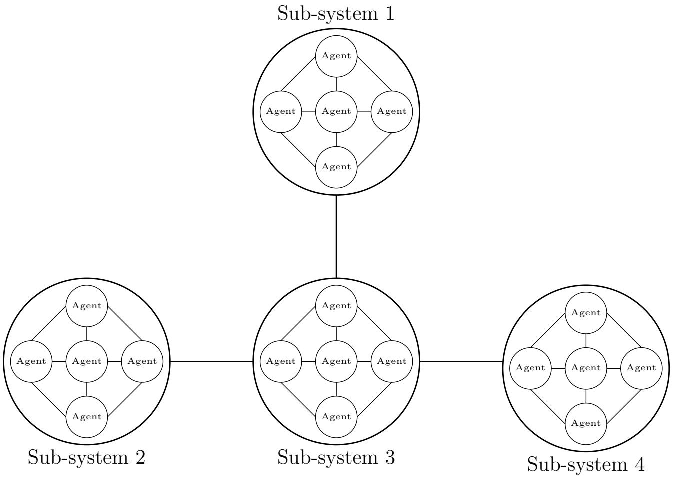


Figure 4.3: Example of a possible configuration of multiple networks interconnected and sharing information.

two strategies. This coupuling interaction is described as follows:

- The agents in the MFG seek an agreement in the stored volume of water inside the sewer pipes.
- Once the best possible agreement has been reached, all agents as a group follow a certain storage element state of the MPC part, e.g., a collector volume or a tank volume.
- This will naturally decrease the volume stored in the pipes and increse the volume stored in the MPC part.
- While the sewage is stored in the MPC portion, the MPC controller transports the water safetly to the WWTP.

Notice that the second step in the interaction above sends water to the “MPC section” only when it is able to process it.

These interactions can be captured by the following cost fuction:

$$J_i^{MP-MFG} = \int_0^T (v_i - \frac{(w_1\bar{m} + w_2 v_{MPC})}{2})^2 + r(q_{in}^i)^2\, dt, \tag{4.8}$$

where $v_{MPC} \in \mathbb{R}$ is the storage variable from the MPC portion, and $w_1, w_2 \in \mathbb{R}$ are tuning parameters. Notice that this function explicitly states that the agents from the MFG are tracking two variables, one asociated with local interactions, and one with global interactions. Notice that since v_{MPC} is a given value to the MFG, cost functions (4.6) and (4.8) have the same structure.

4.2 Problem Statement

Consider a MP-MFG with two populations, where the agents of the first population evolves according to (4.3) and each agent is minimizing the cost functional (4.8). The second population is an MPC-based controller where the control-oriented model (COM) evolves according to a HLD representation as

$$\sum_{i=0}^{T} M_i X(t-i) = m(t), \tag{4.9}$$

$$\sum_{i=0}^{T} N_i X(t-i) \leq n(t), \tag{4.10}$$

where the definitions of all the parameters are given in [JD14], and the controller is trying to maximize the usage of a WWTP (which is a state represented in the vector X) over a known prediction horizon. The problem is to find a set of inflows to the pipes of the MFG population $\{u_1^*, u_2^*, \dots, u_N^*\}$ such that the inequalities

$$J_i^{MP-MFG}(u_1^*, u_2^*, \dots, u_i^*, \dots, u_N^*) \leq J_i^{MP-MFG}(u_1^*, u_2^*, \dots, u_i, \dots, u_N^*) \tag{4.11}$$

are simultaneously satisfied, while at the same time, the flow through retention and redirection gates in the MPC-based population are maximazing the usage of a WWTP.

4.2.1 Controller Implementation

The MP-MFG problem formulation presented previously is stated as a continuous time problem. However, the HLD-based MPC use for the management of the WWTP usage (as presented in [JD14]) is formulated in discrete time. This means that some modifications must be performed to either of the approaches to have a succesful coupling between them. Since the MPC approach requires an online optimization at each sampling time, a continuous-time approach is not convenient. Hence, a discrete-time approach to the MFG is proposed, so that a proper coupling can be achieved. The discretization of the approach is based on the formulation presented in

[NCMH13], where the MFG is presented as a set of coupled differential equations, instead of the canonical coupled non-linear partial differential equations from [LL07], i.e., the Hamilton-Jacobi-Bellman Fokker-Planck-Kolmogorov system. This simplification is possible due to the linear-quadractic (LQ) nature of the game, where the state equation of the agents is linear and the cost function is quadratic.

For the sake of simplicity, the explicit solution of the game is first enunciated. Following the definitions from [NCMH13], it is possible to write an explicit solution to the MP-MFG using traditional tools. These solutions require the implementation of a Riccati equation coupled with an auxilary equation, to compensate for the mean field effect. Notice that this approach is not different from a traditional LQ tracker [LS95]. It is convenient to rewrite the optimization problem for each agent with some auxilary variables, i.e.,

$$\min_{q_{in}^i} \int_0^\infty e^{-\rho t} \left[(v_i - g(\bar{m}))^2 + r(q_{in}^i)^2 \right] dt, \tag{4.12}$$

$$subject\ to$$

$$\dot{v}_i(t) = q_{in}^i - q_{out}^i, \tag{4.13}$$

where $g(\bar{m}) = \frac{(w_1 \bar{m} + w_2 v_{MPC})}{2}$, and the exponential is to accomodate for the infinite-time horizon. As it has been stated in [NCMH13], the previous optimal control problem can be solved by means of the following equations:

$$q_{in}^i = -\frac{1}{r}(p v_i + s), \tag{4.14}$$

$$p^2 + r\rho p - r = 0, \tag{4.15}$$

$$\dot{s}(t) = (\rho + \frac{p}{r}) s(t) + g(\bar{m}), \tag{4.16}$$

$$\dot{\bar{m}}(t) = -\frac{1}{r}(p\,\bar{m}(t) + s), \tag{4.17}$$

where (4.14) is the control law, (4.15) and (4.16) are the Riccati equation and the auxilary equations, and (4.17) is the equation that determines the evolution of the mean field. Equation (4.17) is determined by averaging the state equation of each agent, after applying the control law (4.14). It is important to point out that this solution is consistent with the scheme presented in [LL07], since (4.15) and (4.16) represent the HJB equation, and (4.17) represents the FPK equation.

Given that the previous system of equations is in fact a set of ordinary differential equations, it can be discretizied by means of any discretization tool. For this thesis, the system is discretizied by means of an Euler forward approximation, using a Δt equal to the sampling time from the HLD-based MPC. Therefore, the implementation of the control scheme requires the solution to the following set of equations

$$v_i^+ = v_i + \Delta t(q_{in}^i - q_{out}^i) \tag{4.18}$$

$$s^+ = s + \Delta t\big[(\rho + \frac{p}{r})s + g(\bar{m})\big] \tag{4.19}$$

$$\bar{m}^+ = \bar{m} + \Delta t\big[-\frac{1}{r}(p\,\bar{m} + s)\big] \tag{4.20}$$

where now every variable is in discrete time, and $v_i^+ = v_i(k+1),\ \forall k \in \mathbb{Z}$. Now, the MFG has been discretized, and it is possible to implement it side-by-side with the MPC approach, where v_{MPC} is a simple constant to the MFG.

4.2.2 Constraint Satisfaction Problem

As it has been stated before in this chapter, it is assumed that the inflow to each sewer pipe is controllable. Obviously, this is not a realistic case, since a typical CSN may only have a few gates but hundreds of sewer pipes. For that matter, a map between the optimal inflows computed by the MFG and the available gates must be consider, in order for the implementation to be possible. The map between this two sets of variables is performed by means of a constraint satisfaction problem (CSP), which uses the optimal inflows as a given set. The CSP finds a set of variables such that the optimal inflows provided by the MFG satisfy a set of equations provided by a HLD model.

In the HLD proposed in [JD14], the state vector collects all the important variables from the network such as flows through weirs, overflows on nodes, gate inflows, and inflows to sewer pipes. Hence, the CSP must find a set of every other variable, different from the inflows, such that the model holds. Hence, by using the HLD model, the CSP can be written as follows:

$$\min_{gates} 0$$
$$\textstyle\sum_{i=0}^{T} M_i X(t-i) = m(t),$$
$$\textstyle\sum_{i=0}^{T} N_i X(t-i) \leq n(t),$$

where the only known variables in the state vector X are the inflows to the pipes. Notice that, by finding a vector X that satisfies all the equalities and inequalities above, the inflows running through the gates can be easily read from that vector.

4.2.3 State Estimation Problem

In order to determine their best strategies, agents use the current value of the distribution of others. This means that, in order to compute the optimal strategies, it is required to have full information about the volumes of the pipes. For a real life application, having a sensor at each sewer pipe is quite expensive, and thus, is it possible to measure at a few number of pipes only. For that matter, a way to determine the volume of water inside each pipe, only using the available measured information, should be considered.

For this thesis, it is proposed to use a state observer based on the one proposed on [JD14], which estimates the current state of the network by means of a optimization problem over a moving window, i.e., a moving horizon estimator (MHE). This scheme uses the same HLD model that is used for the CSP that determines the setting for the gates in the controller implementation. The MHE uses information from H_o past steps, and minimizes the diference between the measured outputs and the estimated outputs. This problem is equivalent to a predictive controller, but backwards in nature. Using the HLD, the past information from the network can be written as follows:

$$\sum_{i=0}^{T} M_i X_o(t-i+k) = m(t+k), \tag{4.21}$$

$$\sum_{i=0}^{T} N_i X_o(t-i+k) \leq n(t+k), \tag{4.22}$$

$$k = -H_o + T + 1, \ldots, 0,$$

where H_o is the number of past measured variables that will be used in the problem. Following the propositions from [JD14], the measurments from the system can be written as projections from the state vector as

$$Y(t) = \pi_y X(t),$$

$$U(t) = \pi_U X(t),$$

and the optimization problem associated with the MHE can be written as follows [JD14]:

$$\begin{aligned}
&\min_{X_o,\epsilon_y,\epsilon_U} \mathbf{1}_y^\top \epsilon_y + \mathbf{1}_U^\top \epsilon_U \\
&subject\ to \\
&M_1^o(t)\mathcal{X}_o = M_2^o(t), \\
&N_1^o(t)\mathcal{X}_o \leq N_2^o(t), \\
&-\epsilon_y \leq \Pi_y \mathcal{X}_o - \hat{\mathcal{Y}}(t) \leq \epsilon_y, \\
&-\epsilon_U \leq \Pi_U \mathcal{X}_o - \hat{\mathcal{U}}(t) \leq \epsilon_U, \\
&A_{eq}\mathcal{X}_o(t) = b_{eq}(t), \\
&A_{ineq}\mathcal{X}_o(t) \leq b_{ineq}(t),
\end{aligned}$$

where $\mathcal{X}_o$ is the state vector at all $H_o + T + 1$ time instant, M_1^o, N_1^o, M_2^o and N_2^o are matrices to accomodate for the estimations at all past time instants, $\hat{\mathcal{Y}}(t)$ and $\hat{\mathcal{U}}(t)$ are the measured values of the input and output variables, $\mathbf{1}_y$ and $\mathbf{1}_U$ are vectors of ones of dimensions $H_o \cdot number - of - outputs$ and $H_o \cdot number - of - inputs$ respectively, and ϵ_y and ϵ_U are auxilary variables used to reformulate the minimization of the 1-norms $||\Pi_y \mathcal{X}_o - \hat{\mathcal{Y}}||_1$ and $||\Pi_U \mathcal{X}_o - \hat{\mathcal{U}}||_1$ as a mixed integer linear problem (MILP). Additional equality and inequality constraint are just regular bounds from the state vector and its hybrid nature.

The closed-loop scheme proposed for this thesis is shown in Figure 4.4. It shows all the inputs and outputs from all the important elements from the framework.

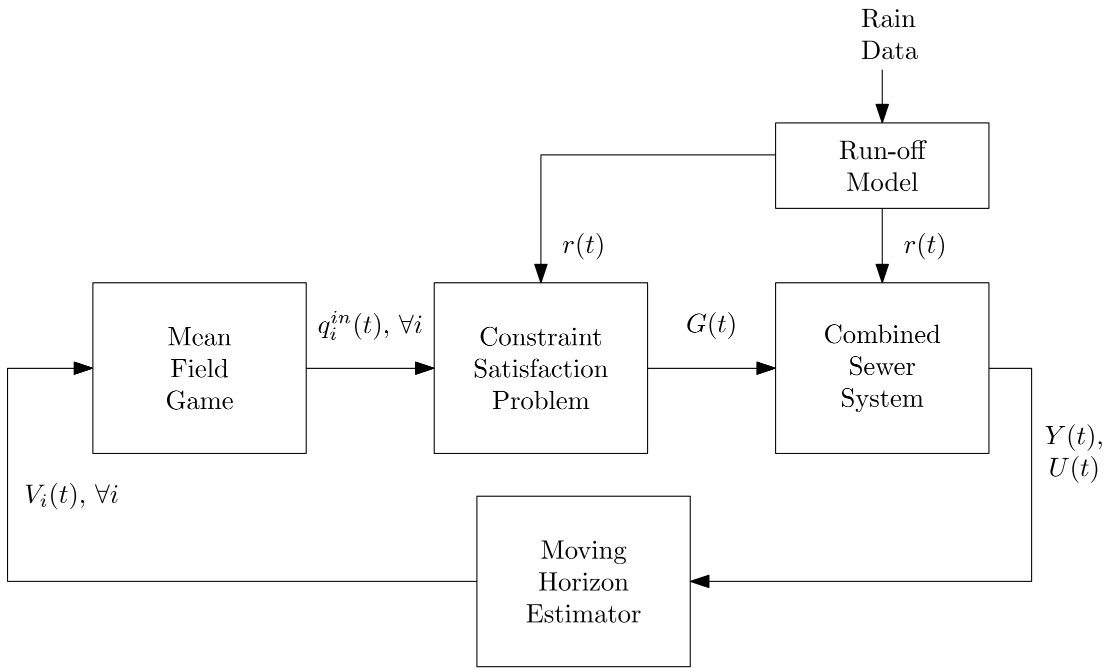


Figure 4.4: Block diagram from the proposed scheme. It shows all the inputs and outputs from all the important elements from the approach.

Part II

Case Study, Results and Concluding Remarks

CHAPTER 5

CASE STUDY AND RESULTS

The proposed controller is tested with the network shown in Figure 3.2a. This UDS is composed of 4 sub-catchments that drain into a tree-like network that, in turn, converges into a common outlet node. This network gives a convenient representation of how a full-size UDS would look like, because of its strong convergence topology. Moreover, this network allows to study one of the most common problem associated with UDSs, which is the uneven use of the pipes of the system, which leads into poor wastewater management, and in most cases, flooding. Hence, this network is a suitable testbed for determining the performance of controllers of UDSs. This network is implemented as a virtual reality programed in DHI MOUSE, and thus all the presented results use the real SVE as part of their core numerical implementation. Hence, the results show how the controller would behave in a real-life implementation.

5.1 Case Study

The proposed scheme is applied into the Riera Blanca network in the city of Barcelona, Spain. The network is shown in Figure 5.1. This is a typical UDS that drains into the Mediterranean sea and a WWTP located downstream of the network. As many UDSs, this network is a collection of several elements such as pipes, tanks, and weirs, that carry the sewage throughout the city. Table 5.1 shows a summary of all the major elements found in this system. As with most UDSs, this system has a quite strong convergence topology, in which the whole system ultimately converges to a single big sewer pipe. This sewer pipes is a large controllable collector that spans over 1.5km and has a quite little slope, causing it to be a suitable storage element. Following the controllable collector downstream, there are the two outlets of the network: the WWTP and the

Mediterranean sea. This WWTP has a maximum capacity of 2 m^3/s, causing any inflow greater than this value to become CSO automatically.

Table 5.1: Parameters of Riera Blanca network

Element	Quantity
Tank	2
Pipe	145
Weir	3
Gate	10
Overflow	11
Collector	1
Rain Inflow	68

The proposed approach utilizes a partition of the system in which one portion is performing the flooding minimization task and the other one is managing the WWTP uses. Given that the outlets of the network are located after the collector and it is able to act as a storage element, it is a suitable idea to divide the network at that point, having both outlets and the collector in the same partition. The selected partitioning allows to decentralize both objectives, as it has been proposed before in this thesis. Figure 5.2 shows the two partitions where the MFG portion is performing the flooding minimization task, while the MPC portion is performing the WWTP usage task. Notice that for this approach, the selected v_{MPC} from (4.8) is the volume from the big sewer pipe. This means that the MFG portion only sends water as long as there is available space inside the big sewer pipe. As for the controller implementation, the system has a sampling time $\Delta t = 1\text{min}$, which is use for both the MP-MFG approach as well as for the MPC approach.

All the information regarding the network is provided by CLABSA (Clavegueram de Barcelona S.A), which includes three-dimensional coordinates of sewer pipes and junctions, crosssectional geometries and materials of sewer pipes, tank geometries and gate characteristics. In order order to test the controllers, a virtual reality of the network is programmed using a DHI's MOUSE callibrated with real data provided by CLABSA, as proposed in [JD14]. This is a quite accurate model that is useful for simulation purposes, since it is capable to model each element of the system, as well as all the switching phenomena from hybrid elements such as weirs and flooding-runoff.

This network has 10 gates that operate as active elements for the system. However, the proposed scheme requires 1 gate for each sewer pipe in the system, thus it would require 145

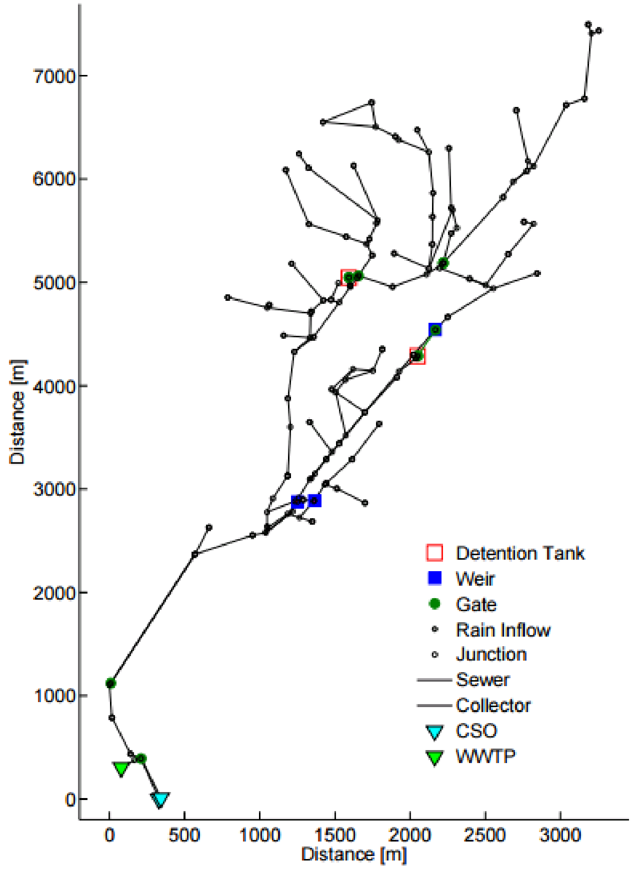


Figure 5.1: Riera Blanca network, Barcelona, Spain.

gates to propertly function. Having that amount of gates is not possible and an additional tool is required to deal with that problem. The main outputs of the MFG portion of the approach are all the inflows to each sewer pipe, knowing that there is a constraint on the maximum inflow. Hence, the MFG portion returns a target value for the flows of the network, which can then be pursued by a local controller at the gates. This task is performed by a quite simple constraint satisfaction problem (CSP) as proposed in [JD14]. This CSP is a follow

$$\min_{gates} 0$$
$$\sum_{i=0}^{T} M_i X(t-i) = m(t),$$
$$\sum_{i=0}^{T} N_i X(t-i) \leq n(t),$$

where all the flows in vector X are already given. Notice that solving this CSP also regulates the

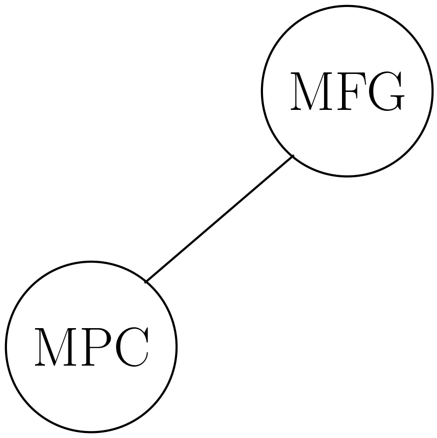


Figure 5.2: Proposed partitioning of the Riera Blanca network for the MP-MFG.

gates that run into the tanks of the network, which ultimately regulates their volumes. Notice that the only desired information from the CSP are the gate flows, which are then pluged into the programmed virtual reality of the network.

5.2 Results and Discussion

The network is tested using three different scenarios: no controller within the loop, full HLD-based MPC, and the proposed MP-MFG approach. These scenarios allow to show the main problems found in the network, as well as the performance and effectiveness of the proposed scheme compared to a more traditional technique. Each scenario is tested using four different real-rainfall events provided by CLABSA from years 2002, 2006, and 2011. Figure 5.3 shows the total inflow entering the network during the four rain events. Notice that each rainfall event has a very distinctive charactistic, which makes them suitable as an impartial benchmarks for simulation. Also, it is important to point out that all the cases are implemented in DHI's MOUSE, and thus they represent the reality as close as possible (as it solves the SVEs).

As it has been stated before, the two main potential problems from this network are the heavy flooding and the poor WWTP usage. Thus, all the plots and results are based upon that data, and no other information is shown unless required.

Figure 5.4 shows the total overflows coming out of the network into street level for all the

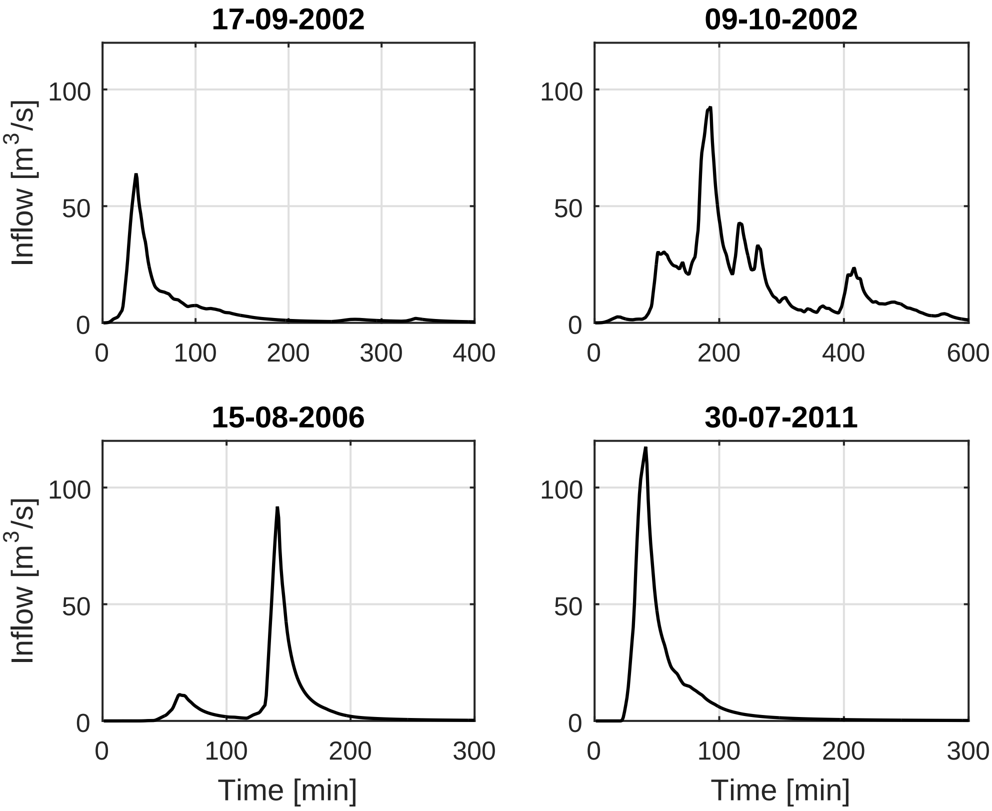


Figure 5.3: Rain-rain scenarios provided by CLABSA used for testing the proposed scheme in the Riera Blanca network.

different control scenarios, and for all the different rain events. When no controller is used in the system, the system presents a serious flooding problem due to the poor management in the active elements. It can be seen that both control strategies, i.e., the MPC and the MP-MFG, are able to reduce the total overflow that the network originally had. It is interesting to see that for the 09-10-2002 scenario, the MP-MFG is not able to do such a good job (compared to the MPC). This is due to the fact that this rain event in not uniform as the others (see Figure 5.3), which causes the mean of the volumes to change quite rapidly, which ultimately misleads the controller. The values of the total volume of overflow are presented in Table 5.2.

Figure 5.5 shows the total inflow entering the WWTP for all the different control scenarios, and for all the different rain events. When no controller is used in the system, the total inflow to the WWTP completely surpasses the maximum capacity of the plant, and thus it instantly becomes CSO. However, when any of the controller schemes are applied, the total flow running

Table 5.2: Total overflow for each scenario

Rain Event	OL [m^3]	MPC [m^3]	MP-MFG [m^3]
17-09-2002	3.7094×10^3	11.5870	2.7496
09-10-2002	2.5752×10^4	176.5420	8.8559×10^3
15-08-2006	6.9475×10^3	22.3741	14.4736
30-07-2011	1.8442×10^4	166.3861	748.9536

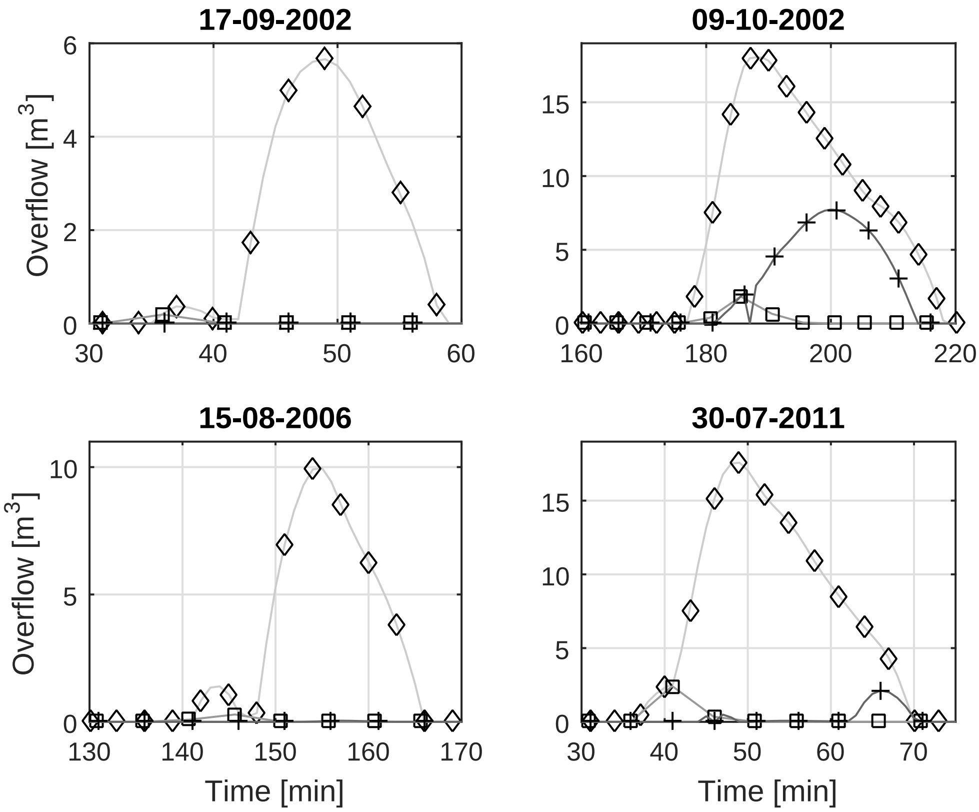


Figure 5.4: Total overflow coming out of the network for all the proposed scenarios, and for the different rain events. The open-loop (OL) overflows are in blue in all graphs, MPC overflows are in red in all graphs, and MP-MFG overflows are in yellow in all graphs.

into the plant stays within its maximum capacity and no sewage is directly sent into the mediterranean sea. It is interesting to notice that for all cases, the MP-MFG takes longer to reach the

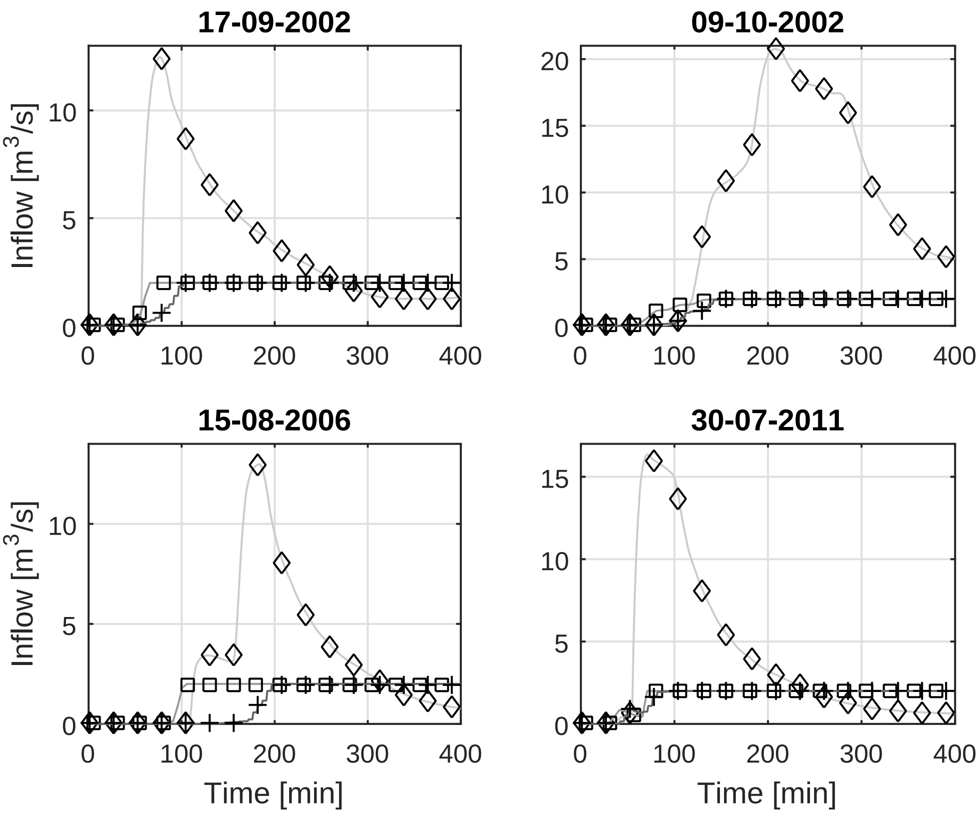


Figure 5.5: Total inflow entering the WWTP for all the proposed scenarios, and for the different rain events. The open-loop (OL) inflows are in blue in all graphs, MPC inflows are in red in all graphs, and MP-MFG inflows are in yellow in all graphs.

maximum capacity of the WWTP. This is due to retention property from the microscopic interaction inside the MFG portion of the scheme. Since pipes seek an agreement on their volumes, it is less important to send water downstream.

Both approaches are able to fullfil the requirements of the system, and guaranteeing a suitable operation. Nonetheless, the MPC approach performes slightly better than the MP-MFG approach. However, this improvement in performance causes the MPC approach to take longer computation times, compared to the MP-MFG approach. From all the rain events presented, the most complex, computationalwise is the 15-08-2006, due to the double peak found in the rain gauge. For this rain event, the MPC approach takes an average of 2.1 time units to compute the solution, while the MP-MFG approach takes 1 time unit, making it faster. This is particulary useful in real-time appliacations where the computational times are an important decision factor.

CHAPTER 6

CONCLUDING REMARKS

In conclusion, this thesis has proposed a control scheme that uses a combination between MFGs and MPC, which allows to successfully control a CSN. The proposed scheme has the advantages from game theoretic approaches, e.g., the ability to have distributed information patterns for some cases and the reduced computational burdens, as well as the advantages from predictive control approaches, e.g., the ability to consider constraints on the states and inputs, and its multi-input multi-output capabilities. It has been shown that the proposed scheme is able to perform within an acceptable performance margin for a virtual reality that uses the SVEs as its numerical core. Also, a simpler scheme based on pure DGs, that can be used for system that do not have a large number of system variables, has been enunciated. The most important aspects from the proposed scheme, as well as the pure DG scheme are listed below.

- The proposed scheme uses a partition of the system that allows to combine multiple control strategies, such as a MFG and a MPC.
- The MFG partition uses non-centralized information patterns, in which the optimal inflow to each sewer pipe is computed by means of an optimal control problem, that uses the average value of the volume on water inside every other sewer pipe.
- The MPC partition uses a HLD COM that is able to represent the network in a quite high level of detail, without considering complex non-linear equations.
- The HLD model used in the MPC partition allows to optimize over any variable inside the network. This means that it is possible to consider objectives such as the maximization of WWTP usage or mantaining some variables below some safety levels.

- The combination between a MFG approch with a HLD based MPC uses less computational resources compare to a full HLD-based MPC, which in term, derives in less computation times.
- The pure DG proposed approach is able to consider completely distributed information patterns on local controllers of a CSN.
- Since only a DG is consider in the pure DG approach, it runs significally faster compared to a linear MPC approach.
- The main proposed scheme in this thesis has been tested in DHI MOUSE, which shows how the scheme would perform on a real-life implementation for the selected network, i.e., the Riera Blanca network.

6.1 Contributions

The main contributions from this thesis are outlined as follows:

- Improving the computation times for a real-time implementation of controllers for CSN, without sacrificing performance.
- Determining completely distributed schemes, based on consensus-like algorithms, for the control of CSNs.
- Combining two distinct control strategies, which have completely different advantages, deriveng in a versatile framework.
- Implementing the proposed scheme on a real network using a virtual reality and not only a linear model.

6.2 Directions for Future Research

Since this thesis has three distinct topics underlying the main constributions, i.e., DGs, MFGs, and MFGs combined with other approaches, the future directions are quite broad. Here, a few of this directions are listed.

- As for the real implementation of the schemes, in this thesis it was asumed that it is possible to have measures of inflows to sewer pipes. Although this is possible, it is quite expensive. It is a more suitable idea to consider level measures in the pipes, which are gotten from much cheaper sensors.

- As for the MFG approach in the proposed scheme, the game is not able to explicitly consider constraints that the system may have. All constraints were satisfied by using saturations on the input and states variables. This causes a degradation in the performance of the global approach. For that matter, it is recomended to consider a mechanism that allows to consider constraints in a OCP such as a dual MPC approach.

- As for the pure DG approach, it is quite relevant to consider delayed models for the strategies of the agents. This is due to the fact that there are delays on every sewer pipe in the system that the agents should consider.

- Finally, it should be pointed out that the proposed scheme allows to consider any combination of a control strategy with a MFG. Hence, it is possible to study the interaction between a MFG and other strategies, such as a pure DG.

Part III

Appendices

APPENDIX A

ACRONYMS

MPC	Model Predictive Control
MFG	Mean Field Game
MPMFG	Multi Population Mean Field Game
DG	Differential Game
CSN	Combined Sewer Network
UDS	Urban Drainage System
OCP	Optimal Control Problem
MHE	Moving Horizon Estimator
HLD	Hybrid Linear Deleyad
COM	Control Oriented Model
WWTP	Wastewater Treatment Plant
CSP	Constraint Satisfaction Problem
NE	Nash Equilibrium
VT	Virtual Tank
VFC	Volume to Flow Conversion
MILP	Mixed Integer Linear Problem
HJB	Hamilton Jacobi Bellman
FPK	Fokker Planck Kolmogorov
CSO	Combined Sewer Overflows
SVE	Saint Venant Equation

BIBLIOGRAPHY

[BB08] T. Başar and P. Bernhard. *H-∞ optimal control and related minimax design problems: a dynamic game approach*. Springer Science & Business Media; Boston, MA, 2008.

[BD62] Richard Ernest Bellman and Stuart E Dreyfus. *Applied dynamic programming*. Rand Corporation, 1962.

[BD04] D. Butler and J. Davies. *Urban drainage*. Spon Press; London, 2004.

[BGORB$^+$15] J. Barreiro-Gomez, G. Obando, G. Riaño-Briceño, N. Quijano, and C. Ocampo-Martinez. Decentralized control for urban drainage systems via population dynamics: Bogotá case study. In *Proceedings of the European Control Conference (ECC)*, Linz, Austria, 2015.

[BMA14] D. Bauso, T. Mylvaganam, and A. Astolfi. Transient stability of a power grid via robust mean field games. *Proceedings of NetGCoop*, Trento, Italy, 2014.

[BO95] T. Basar and G. Olsder. *Dynamic noncooperative game theory*, volume 200. SIAM; Warrendale, PA, 1995.

[Bre11] Alberto Bressan. Noncooperative differential games. *Milan Journal of Mathematics*, 79(2):357–427, 2011.

[CAS08] A. Cardenas, S. Amin, and S. Sastry. Secure control: Towards survivable cyber-physical systems. In *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops*, pages 495–500, Beijing, China, 2008.

[Cho59] V. T. Chow. *Open channel hydraulics*. McGraw-Hill Book Company, Inc; New York, 1959.

[CQS+04] G. Cembrano, J. Quevedo, M. Salamero, V. Puig, J. Figueras, and J. Martí. Optimal control of urban drainage systems. A case study. *Control Engineering Practice*, 12(1):1–9, 2004.

[CSMndlPnL13] P. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

[Eng05] J. Engwerda. *LQ dynamic optimization and differential games*. John Wiley & Sons; New York, 2005.

[FEMRH15] N. Forouzandehmehr, M. Esmalifalak, H. Mohsenian-Rad, and Z. Han. Autonomous demand response using stochastic differential games. *IEEE Transactions on Smart Grid*, 6(1):291–300, 2015.

[FKV12] P. Falugi, P. A. Kountouriotis, and R. B. Vinter. Differential games controllers that confine a system to a safe region in the state space, with applications to surge tank control. *IEEE Transactions on Automatic Control*, 57(11):2778–2788, 2012.

[GBGE+15] L. García, J. Barreiro-Gomez, E. Escobar, D. Téllez, N. Quijano, and C. Ocampo-Martinez. Modeling and real-time control of urban drainage systems: A review. *Advances in Water Resources*, 85:120 – 132, 2015.

[Gu08] D. Gu. A differential game approach to formation control. *IEEE Transactions on Control Systems Technology*, 16(1):85–93, 2008.

[HCM03] Minyi Huang, Peter E Caines, and Roland P Malhamé. Individual and mass behaviour in large population stochastic wireless power control problems: centralized and Nash equilibrium solutions. In *Proceedings. 42nd IEEE Conference on Decision and Control, 2003.*, volume 1, pages 98–103, Maui, Hawaii, USA, 2003.

[JD14] B. Joseph-Duran. *Hybrid Modelling and Receding Horizon Control of Combined Sewer Networks*. PhD thesis, Universitat Politècnica de Catalunya, 2014.

[JDJOM+14] B. Joseph-Duran, M. N. Jung, C. Ocampo-Martinez, S. Sager, and G. Cembrano. Minimization of sewage network overflow. *Water Resources Management*, 28(1):41–63, 2014.

[Kir12] Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2012.

[LL07] J.M. Lasry and P.L. Lions. Mean field games. *Japanese Journal of Mathematics*, 2(1):229–260, 2007.

[LS95] F. L. Lewis and V. L. Syrmos. *Optimal control*. John Wiley & Sons, 1995.

[Mac02] J. M. Maciejowski. *Predictive control: with constraints*. Pearson education; Englewood Cliffs, NJ, 2002.

[MMDLPC11] J.M. Maestre, D. Muñoz De La Peña, and E.F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32(2):153–176, 2011.

[MSA14] T. Mylvaganam, M. Sassano, and A. Astolfi. A constructive differential game approach to collision avoidance in multi-agent systems. In *Proceedings of the American Control Conference (ACC), 2014*, pages 311–316, Portland, USA, 2014.

[NADZ13] I. V. Nagy, K. Asante-Duah, and I. Zsuffa. *Hydrological dimensioning and operation of reservoirs: practical design concepts and principles*, volume 39. Springer Science & Business Media, 2013.

[Nas50] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

[NCMH13] M. Nourian, P. Caines, R. Malhame, and M. Huang. Nash, social and centralized solutions to consensus problems via mean field control theory. *IEEE Transactions on Automatic Control*, 58(3):639–653, 2013.

[OM10] C. Ocampo-Martinez. *Model predictive control of wastewater systems*. Springer Verlag, 2010.

[RBRJBG+15] G. Riaño-Briceño, A. Ramirez-Jaime, J. Barreiro-Gomez, N. Quijano, and C. Ocampo-Martinez. Co-simulation for the design of controllers in urban drainage systems. In *Proceedings of the IEEE Colombian Conference on Automatic Control (CCAC)*, Manizales, Colombia, 2015.

[WANON12] P. Willems, K. Arnbjerg-Nielsen, J. Olsson, and V. Nguyen. Climate change impact assessment on urban rainfall extremes and urban drainage: methods and shortcomings. *Atmospheric research*, 103:106–118, 2012.